CORPORATE INFORMATION MANAGEMENT

THE EXECUTIVE LEVEL GROUP

OCTOBER 15, 1990

CORPORATE INFORMATION MANAGEMENT

FOCUS ON: PROCESSES AND PROCEDURES

Presented to

THE EXECUTIVE LEVEL GROUP

Mrs. Belkis Leong-Hong
Director, Corporate Information Management

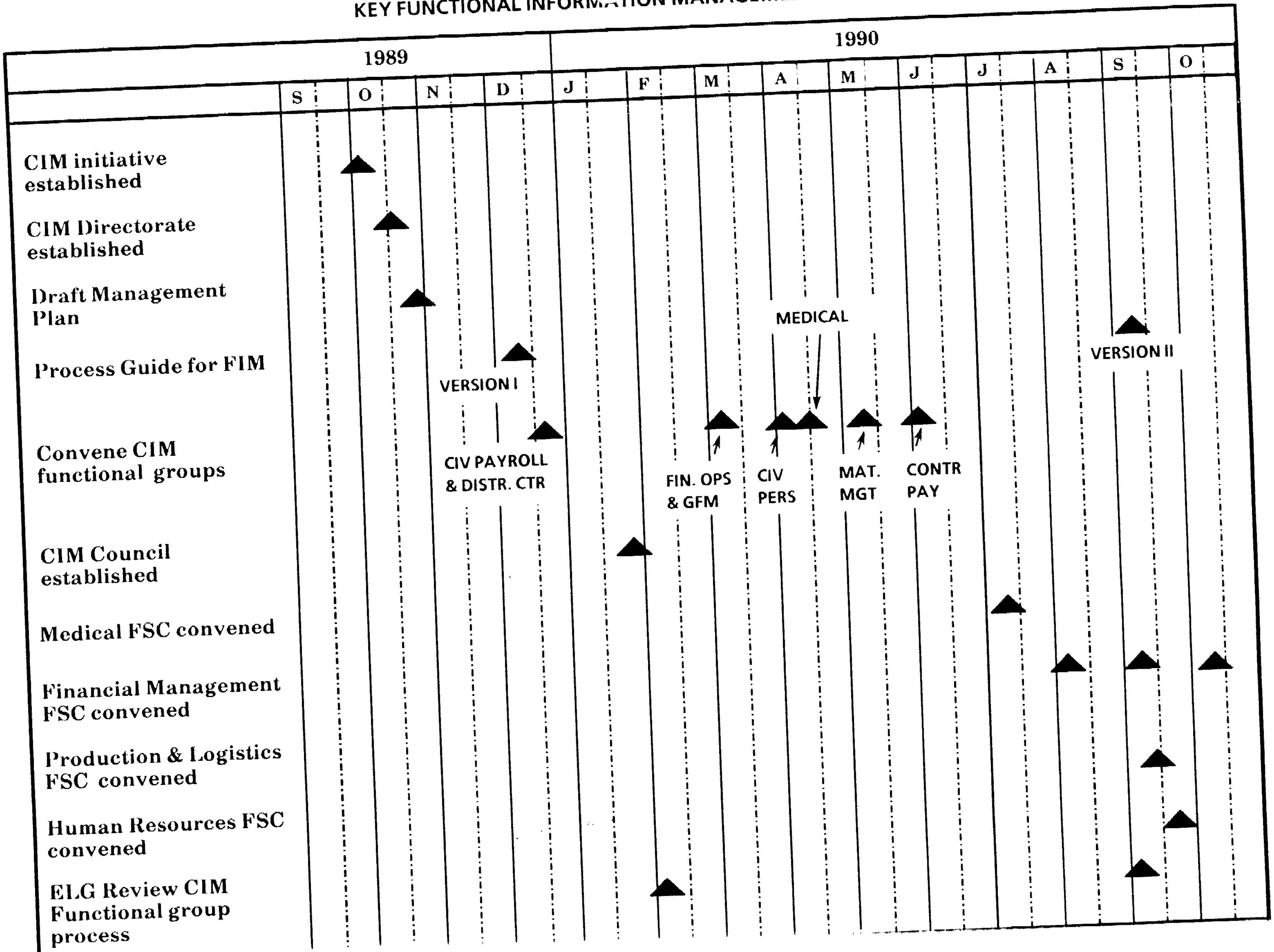
October 15, 1990

TASKING FROM

OCTOBER 4, 1989 MEMORANDUM

- IRM STAFF WILL :
  - DRAFT A MANAGEMENT PLAN
  - DEVELOP A PROCESS GUIDE
  - ESTABLISH FUNCTIONAL GROUPS

KEY FUNCTIONAL INFORMATION MANAGEMENT EVENTS



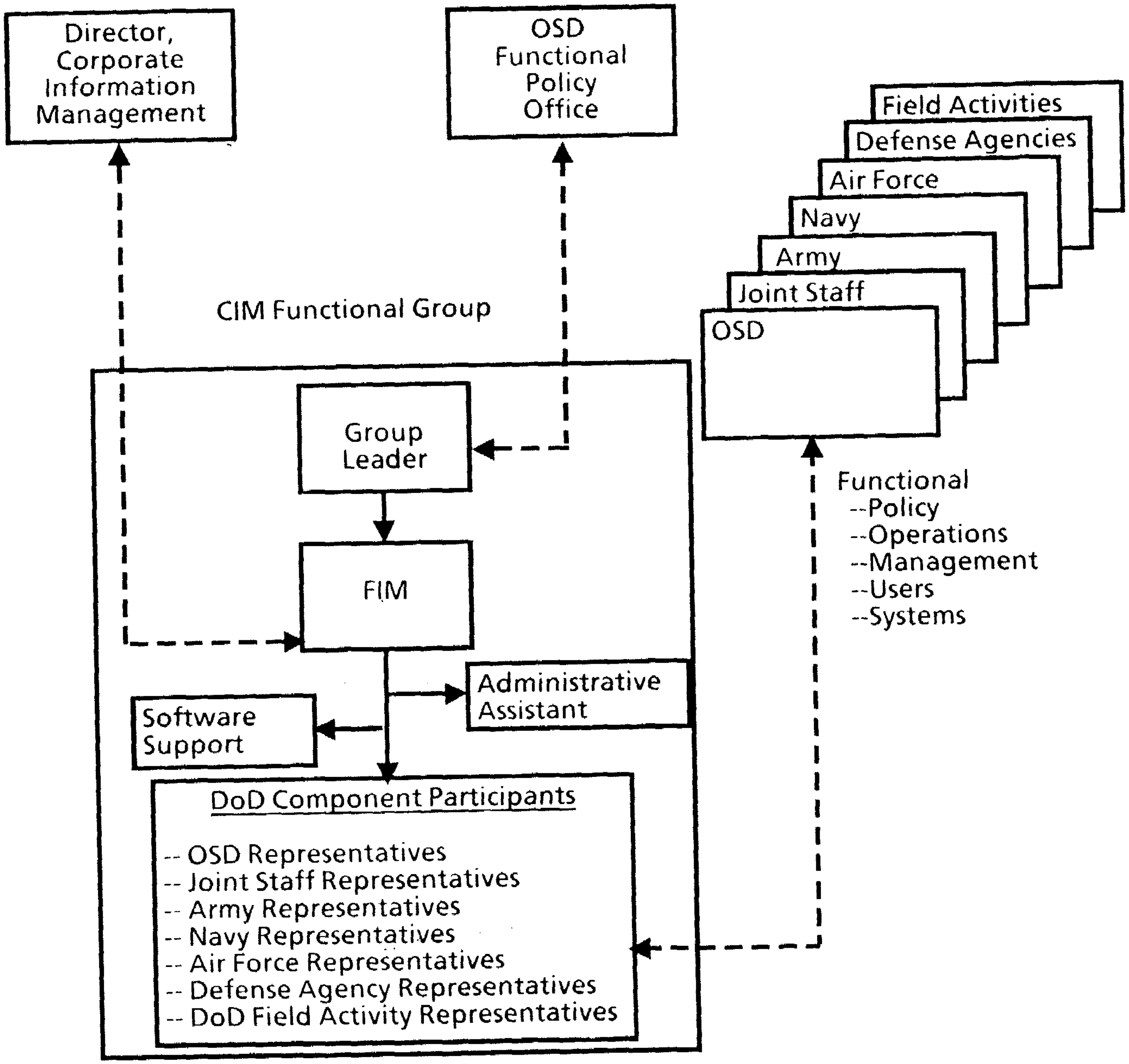
GUIDING PRINCIPLES

- BUSINESS METHOD IS THE DOMAIN OF THE USER COMMUNITY, THEREFORE, FUNCTIONAL LEADERSHIP AND PARTICIPATION IN THE FUNCTIONAL GROUPS ARE ESSENTIAL AND CRITICAL TO THE SUCCESS OF FUNCTIONAL INFORMATION MANAGEMENT
- BUSINESS METHODS AND CURRENT PRACTICES ASSESSMENT ARE THE BASIS FOR RECOMMENDING CHANGES TO ACHIEVE BETTER, MORE EFFICIENT, AND MORE EFFECTIVE WAYS OF DOING BUSINESS; INNOVATIVE FUNCTIONAL PROCESSES FRAME FUNCTIONAL DATA AND PROCESS MODELS
- ACT AS A CATALYST FOR THE FUNCTIONAL COMMUNITY TO EVALUATE AND ADOPT NEW AND BETTER WAYS OF DOING BUSINESS
- WELL-DEFINED FUNCTIONAL REQUIREMENTS, CONSISTING OF COMMON PROCESS AND DATA MODELS ARE PREREQUISITES TO IMPLEMENTING COMMON INFORMATION SYSTEMS
- INFORMATION SYSTEMS SHOULD ONLY BE DEVELOPED TO SUPPORT FUNCTIONAL NEEDS AND FUNCTIONAL POLICIES
- STANDARD DATA DEFINITION AND A DICTIONARY/REPOSITORY ARE ESSENTIAL TO DEFINING STABLE DATA AND PROCESS MODELS
- EMPHASIS ON COST/BENEFIT AND FEASIBILITY IN PRIORITIZING AND ASSESSING FUNCTIONAL REQUIREMENTS

OVERALL APPROACH FOR IMPLEMENTING FUNCTIONAL INFORMATION MANAGEMENT

- EMPHASIS ON FUNCTIONAL LEADERSHIP AND PARTICIPATION
- LONG-TERM, FULL-TIME PARTICIPATION BY GROUP LEADER AND MEMBERS IN DEVELOPING PRODUCTS FOR EACH FUNCTIONAL AREA.
- FUNCTIONAL GROUPS DOCUMENT BUSINESS METHODS WITHIN FUNCTIONAL AREAS
- FUNCTIONAL GROUPS FOCUS ON:
    - BUSINESS METHODS DEFINITION AND DOCUMENTATION
    - INNOVATIVE FUNCTIONAL VISION TO FRAME FUNCTIONAL DATA AND PROCESS MODELLING
    - DEVELOPING STABLE FUNCTIONAL PROCESS AND DATA MODELS AS A PREREQUISITE FOR DEVELOPING COMMON INFORMATION SYSTEMS
- USE OF CONSISTENT PROCESS AND METHODOLOGY TO DEVELOP FUNCTIONAL REQUIREMENTS FOR A COMMON SYSTEM TO SUPPORT THE FUNCTIONAL AREA

CIMFUNCTIONAL GROUP RELATIONSHIPS



PROCESS OVERVIEW

PROCESS APPROACH

- A STRUCTURED PROCESS FOR:
  - CONSISTENCY AND COMPARABILITY OF OUTPUTS
  - COMMON LANGUAGE ACROSS FUNCTIONAL GROUPS
  - FRAMEWORK FOR DISCUSSIONS AND NEGOTIATIONS
  - PROCESS REVIEW AND QUALITY ASSURANCE
- LINK STRATEGIC PLANNING WITH INFORMATION ENGINEERING, BUSINESS SYSTEMS MODELLING, AND INFORMATION SYSTEMS REQUIREMENTS ANALYSIS
- DIFFERENCES WITH COMMONLY USED APPROACHES
  - FUTURE ORIENTED
  - DUAL PATH (NEITHER PROCESS NOR DATA MODEL DOMINANT)
- SUPPORT THROUGH TOOLS
- DEVELOPED WITH VIEW TO OVERCOMING PAST DESIGN AND DEVELOPMENT PROBLEMS

PROCESS GUIDE OVERVIEW

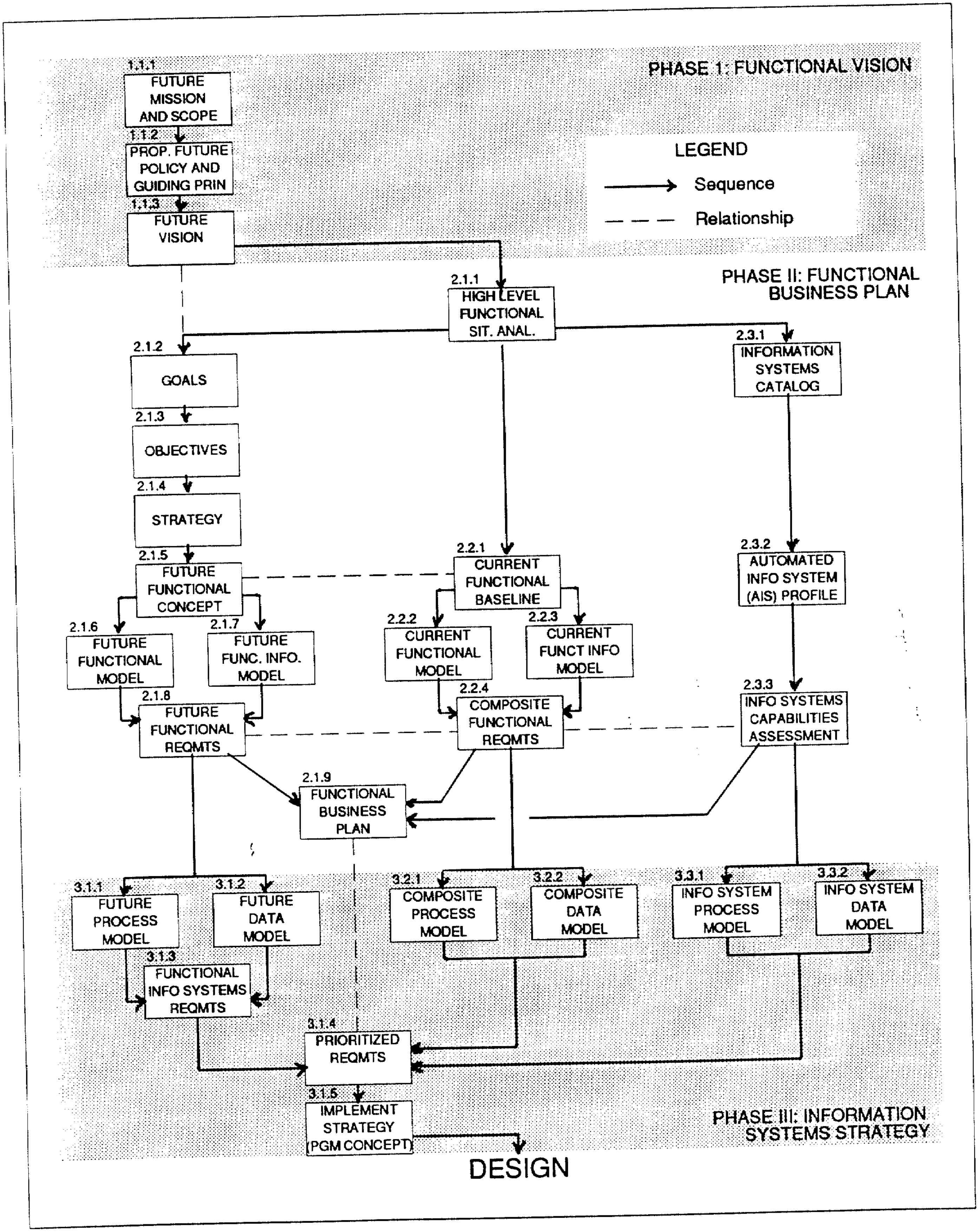
PROCESS GUIDE OVERVIEW

- THREE PHASES FROM HIGH LEVEL STRATEGIC PLANNING THROUGH INFORMATION SYSTEMS REQUIREMENTS (see chart)
- EACH PHASE HAS PRODUCT OUTPUT WHICH IS MAJOR DECISION POINT PRIOR TO NEXT SEGMENT (see chart):
  - <u>FUNCTIONAL VISION</u>: COMBINATION OF BROAD GUIDANCE FROM FUNCTIONAL STEERING COMMITTEE AND THOROUGH ANALYSIS FACILITATED IN THE GROUPS
  - <u>FUNCTIONAL BUSINESS PLAN</u>: COMPOSITE OF ANALYSIS OF FUNCTIONS AND DATA REQUIREMENTS, WITH A BUSINESS CASE FOR EXECUTION, BASED ON ECONOMICS; INTERMEDIATE DECISION POINTS WITH GUIDANCE FROM THE FUNCTIONAL STEERING COMMITTEE
  - <u>IMPLEMENTATION STRATEGY</u>: PROGRAM CONCEPT AND REQUIRED ACTIONS WITH TRANSITION CONCEPT
- EACH PHASE IS INITIATED BY DEVELOPING COMMON UNDERSTANDING OF THE BUSINESS AREA AND THE DIRECTION FOR THE PHASE (AND GREATER LEVELS OF DETAIL)

  -MISSION/SCOPE

  -SITUATION ANALYSIS

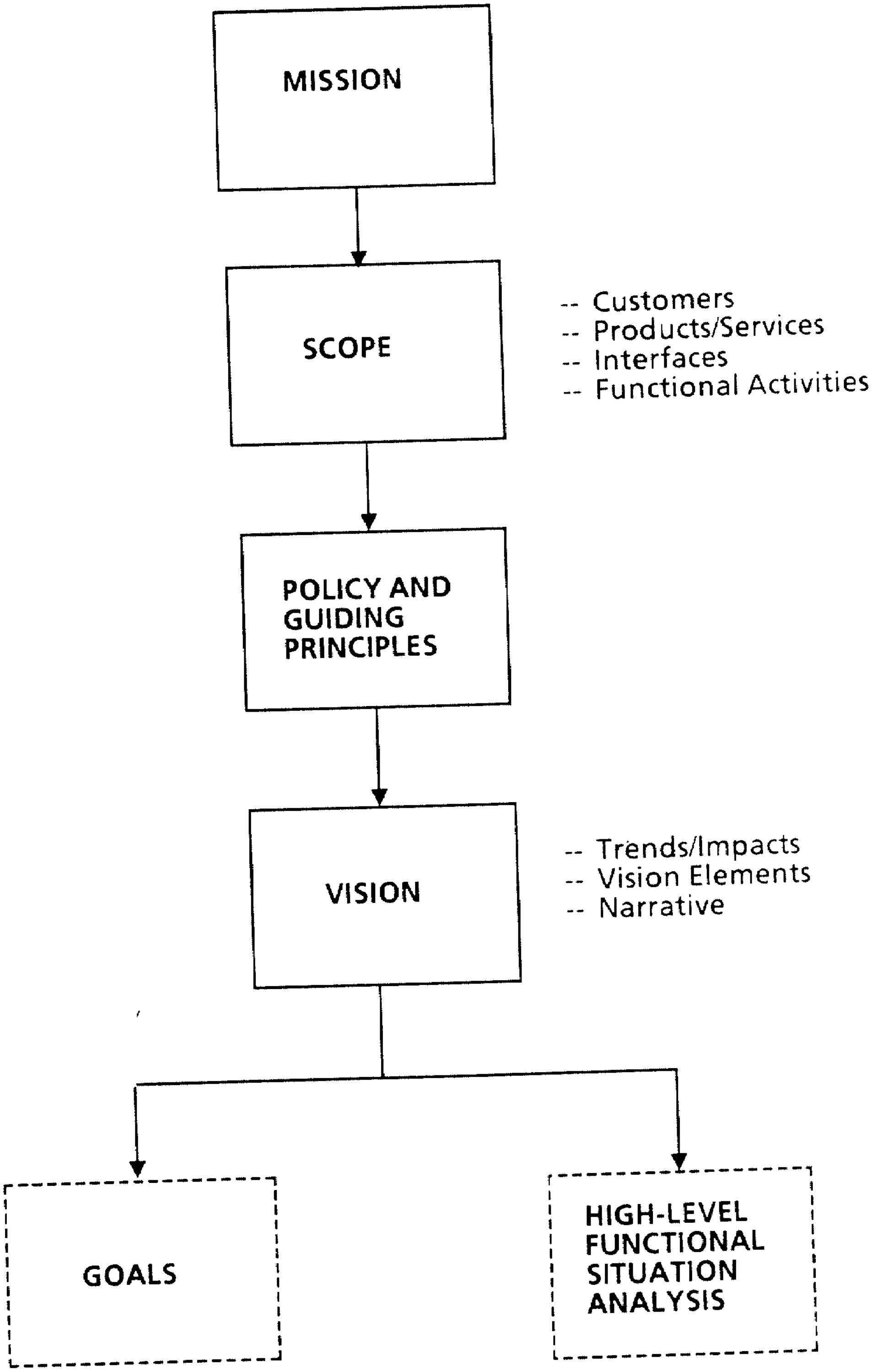
  -FUNCTIONAL PLAN ASSESSMENT


PHASE 1: FUNCTIONAL VISION
1.1.1 FUTURE MISSION AND SCOPE
1.1.2 PROP. FUTURE POLICY AND GUIDING PRIN
1.1.3 FUTURE VISION
LEGEND
Sequence
Relationship
PHASE II: FUNCTIONAL BUSINESS PLAN
2.1.1 HIGH LEVEL FUNCTIONAL SIT. ANAL.
2.1.2 GOALS
2.1.3 OBJECTIVES
2.1.4 STRATEGY
2.1.5 FUTURE FUNCTIONAL CONCEPT
2.1.6 FUTURE FUNCTIONAL MODEL
2.1.7 FUTURE FUNC. INFO. MODEL
2.1.8 FUTURE FUNCTIONAL REQMTS
2.1.9 FUNCTIONAL BUSINESS PLAN
2.2.1 CURRENT FUNCTIONAL BASELINE
2.2.2 CURRENT FUNCTIONAL MODEL
2.2.3 CURRENT FUNCT INFO MODEL
2.2.4 COMPOSITE FUNCTIONAL REQMTS
2.3.1 INFORMATION SYSTEMS CATALOG
2.3.2 AUTOMATED INFO SYSTEM (AIS) PROFILE
2.3.3 INFO SYSTEMS CAPABILITIES ASSESSMENT
3.1.1 FUTURE PROCESS MODEL
3.1.2 FUTURE DATA MODEL
3.1.3 FUNCTIONAL INFO SYSTEMS REQMTS
3.1.4 PRIORITIZED REQMTS
3.1.5 IMPLEMENT STRATEGY (PGM CONCEPT)
3.2.1 COMPOSITE PROCESS MODEL
3.2.2 COMPOSITE DATA MODEL
3.3.1 INFO SYSTEM PROCESS MODEL
3.3.2 INFO SYSTEM DATA MODEL
PHASE III: INFORMATION SYSTEMS STRATEGY
DESIGN

PHASE 1: FUNCTIONAL VISION

- OBJECTIVES
  - TO VISUALIZE THE FUNCTION OF THE FUTURE
  - TO INITIATE "BUY-IN" TO JOINT BUSINESS PRACTICES AND PROCEDURES
- MAJOR TASKS
  - APPLY SENIOR POLICY DIRECTION TO DEFINE GUIDING PRINCIPLES AND VISION
  - DEFINE A FUTURE-ORIENTED AND UNCONSTRAINED MISSION AND VISION
- TOOLS AND TECHNIQUES
  - ANALYSIS OF FUTURES
  - TEAM BUILDING
- MAJOR OUTPUTS
  - VISION STATEMENTS
- RELATIONSHIP TO OTHER PHASES AND TASKS
  - VISION STATEMENTS PROVIDE ARCHITECTURE FOR REST OF ANALYSIS
  - INITIAL IDENTIFICATION OF INTERFACES AND INTEGRATION REQUIREMENTS

PHASE I - FUNCTIONAL VISION



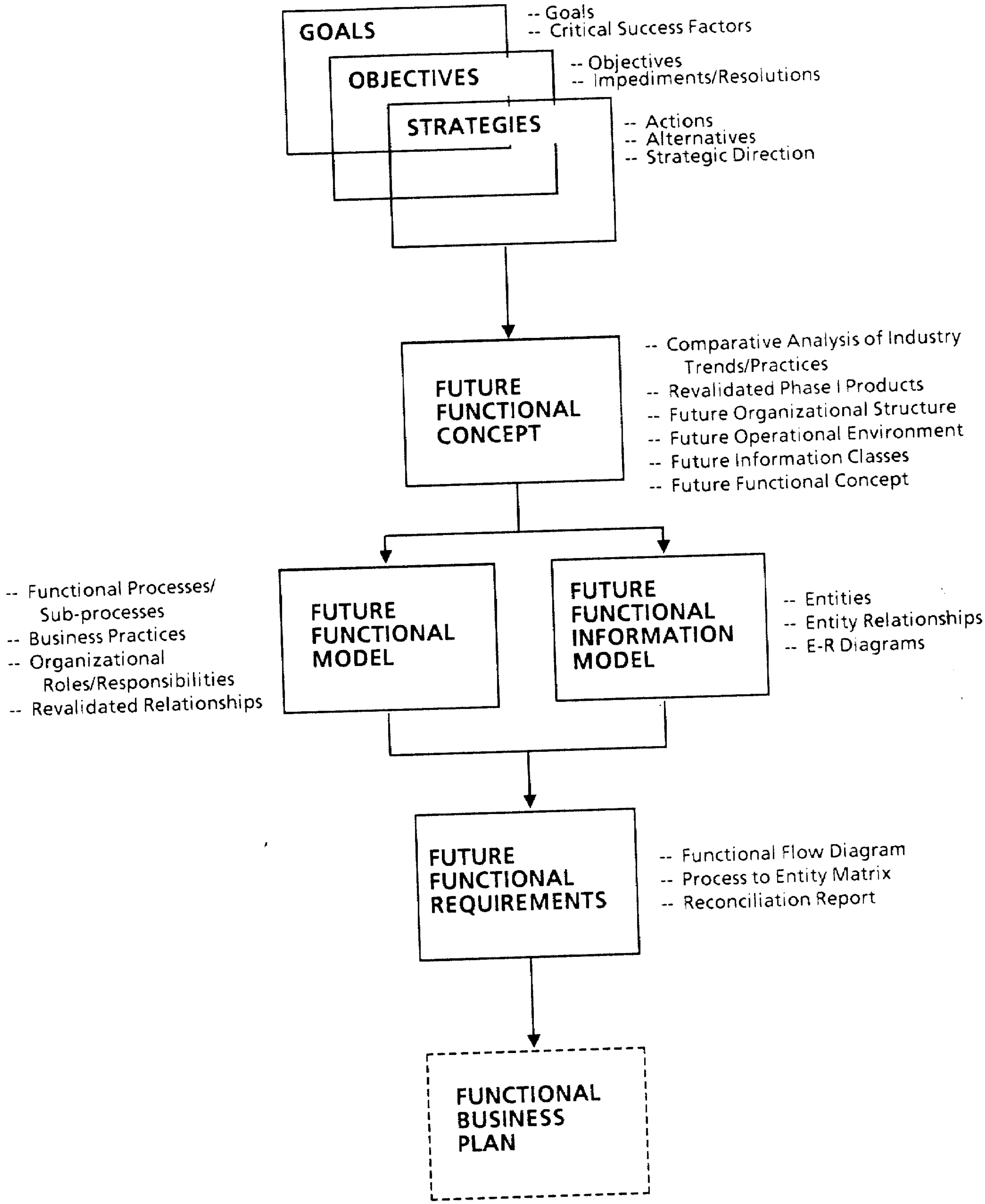
PHASE 2: FUNCTIONAL BUSINESS PLAN

- OBJECTIVES
  - TO DEVELOP A FEASIBLE STRATEGY FOR MEETING THE FUTURE VISION, GIVEN THE CURRENT SITUATION
  - TO MAKE THE BUSINESS CASE FOR PURSUING THE FUTURE CONCEPT
  - TO DOCUMENT CURRENT ENVIRONMENT USING STANDARD LANGUAGE AND CONCEPTS
- MAJOR TASKS
  - DETERMINE STRATEGIES TO MEET VISION
  - DEFINE THE FUTURE AND DOCUMENT THE CURRENT CONCEPT OF OPERATIONS
  - DEFINE FUTURE AND CURRENT FUNCTIONAL REQUIREMENTS
  - ASSESS CURRENT INFORMATION SYSTEMS
- TOOLS AND TECHNIQUES
  - FEASIBILITY AND RISK ASSESSMENTS
  - BUSINESS PLANNING (PROJECT SCHEDULING, BENEFIT ANALYSIS, CRITICAL SUCCESS FACTORS)
  - FUNCTIONAL DECOMPOSITION
  - ENTITY ANALYSIS

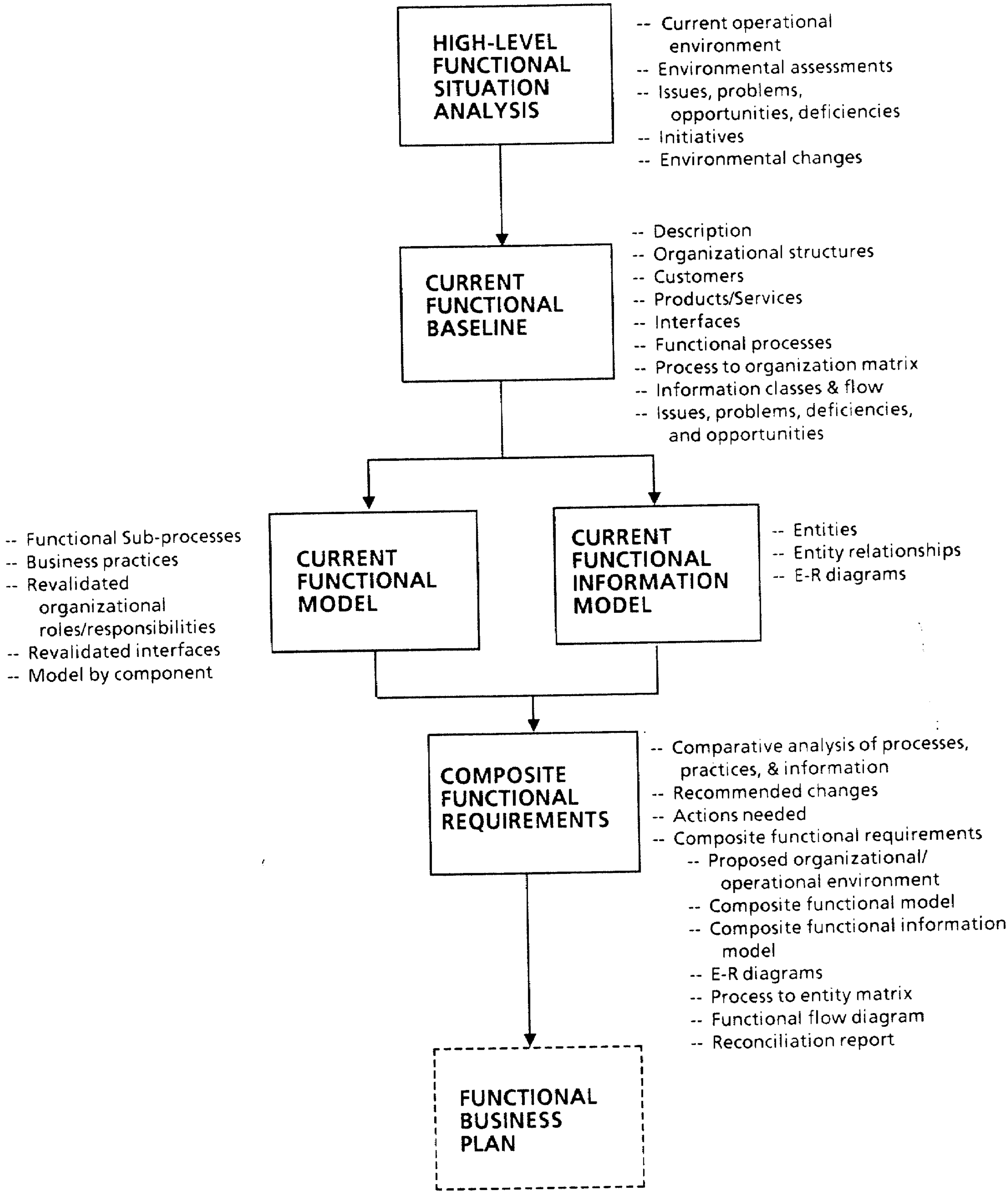
PHASE 2: FUNCTIONAL BUSINESS PLAN (CONT)

- MAJOR OUTPUTS
  - FUTURE FUNCTIONAL REQUIREMENTS
  - JOINT CURRENT FUNCTIONAL REQUIREMENTS
  - CURRENT INFORMATION SYSTEMS ASSESSMENT
  - FUNCTIONAL BUSINESS PLAN
- RELATIONSHIP TO OTHER PHASES AND TASKS
  - VISION ELEMENTS BASIS FOR FUTURE CONCEPTS AND FUNCTIONAL REQUIREMENTS
  - JOINT PRACTICES AND FUNCTIONAL REQUIREMENTS DRIVE INFORMATION SYSTEMS REQUIREMENTS

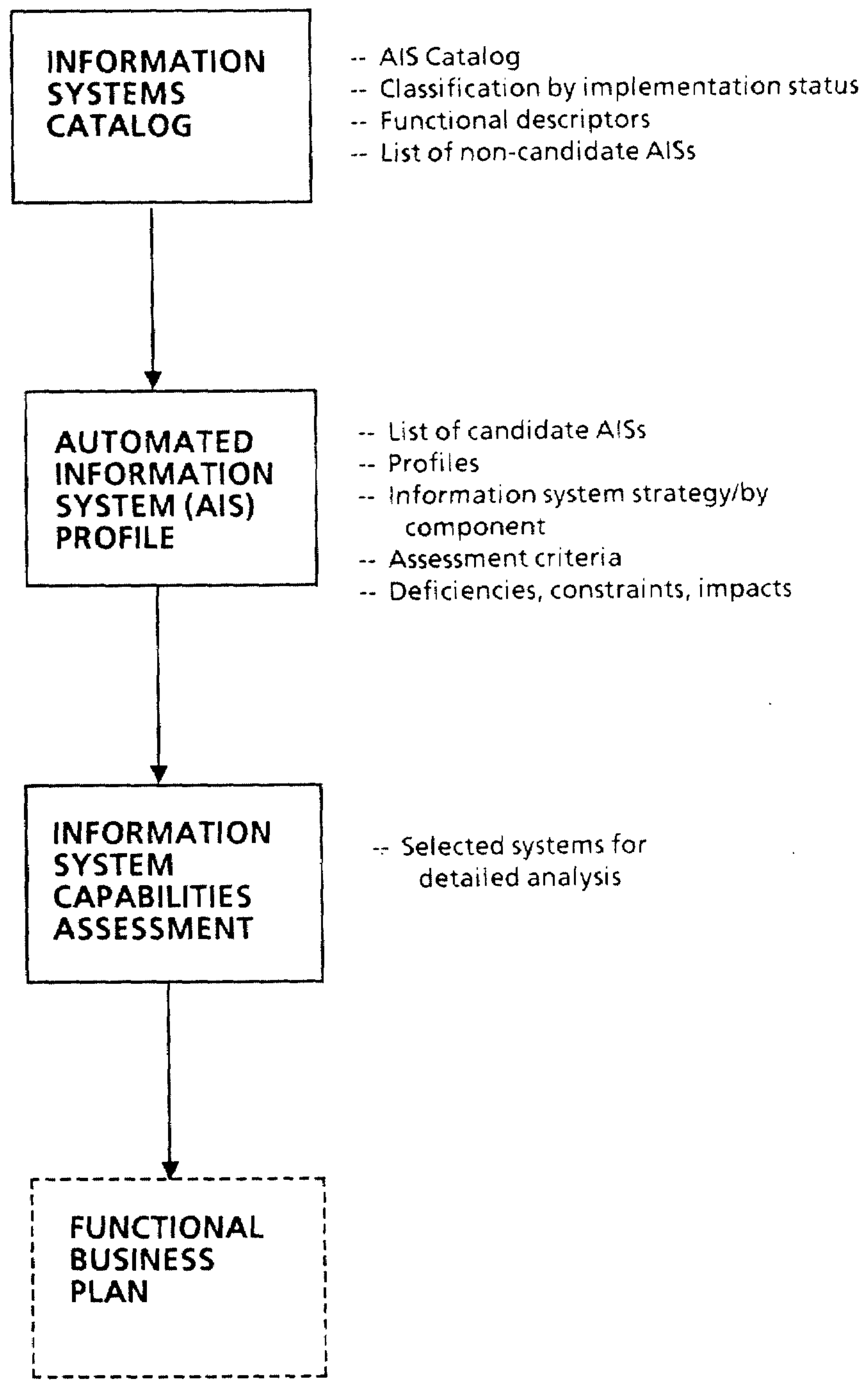
PHASE II - FUNCTIONAL BUSINESS PLAN
(Future Functional Path)



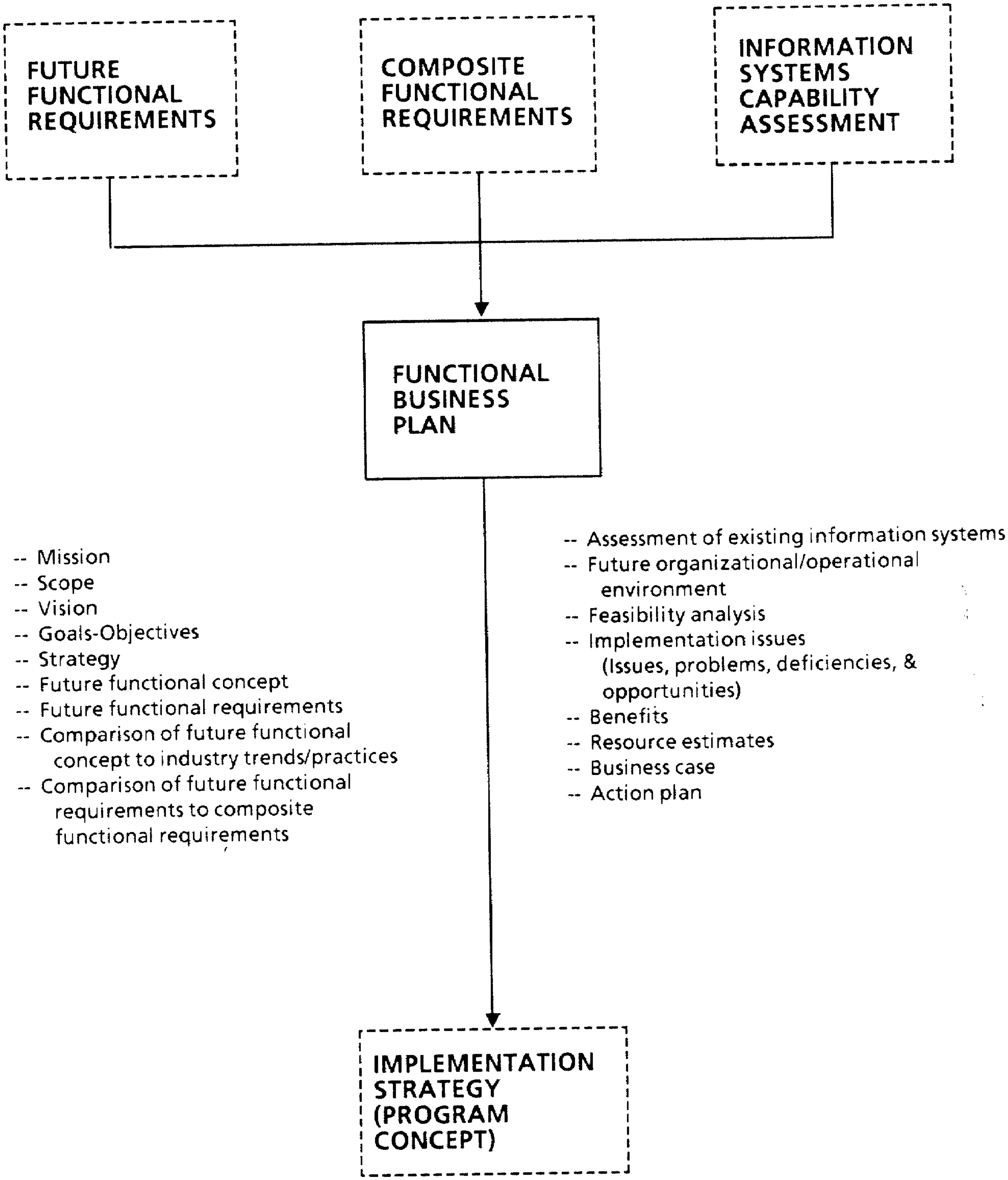
PHASE II - FUNCTIONAL BUSINESS PLAN
(Current Functional Path)



PHASE II - FUNCTIONAL BUSINESS PLAN
(Current Information Systems Path)



PHASE II - FUNCTIONAL BUSINESS PLAN



PHASE 3: INFORMATION SYSTEMS STRATEGY

- OBJECTIVES
  - TO PREPARE INFORMATION SYSTEMS FUNCTIONAL REQUIREMENTS FOR SYSTEMS DESIGN
  - TO DEVELOP A TRANSITION STRATEGY FOR MIGRATING FROM CURRENT TO FUTURE SYSTEMS
- MAJOR TASKS
  - DEFINE STABLE PROCESS AND DATA MODELS FOR NEW INFORMATION SYSTEMS ARCHITECTURE
  - DEVELOP AND PRIORITIZE REQUIREMENTS
  - DEVELOP TRANSITION STRATEGY FROM CURRENT TO FUTURE PROCESS AND DATA MANAGEMENT
  - DEVELOP REALISTIC IMPLEMENTATION STRATEGY BASED ON BEST BUSINESS PRACTICE
- TOOLS AND TECHNIQUES
  - DATA AND PROCESS MODELING
  - PRIORITIZATION BASED ON TECHNICAL, OPERATIONAL, MANAGERIAL AND POLITICAL FACTORS
  - BUSINESS PLANNING (PROJECT SCHEDULING, BENEFIT ANALYSIS, DEPENDENCE ANALYSIS)

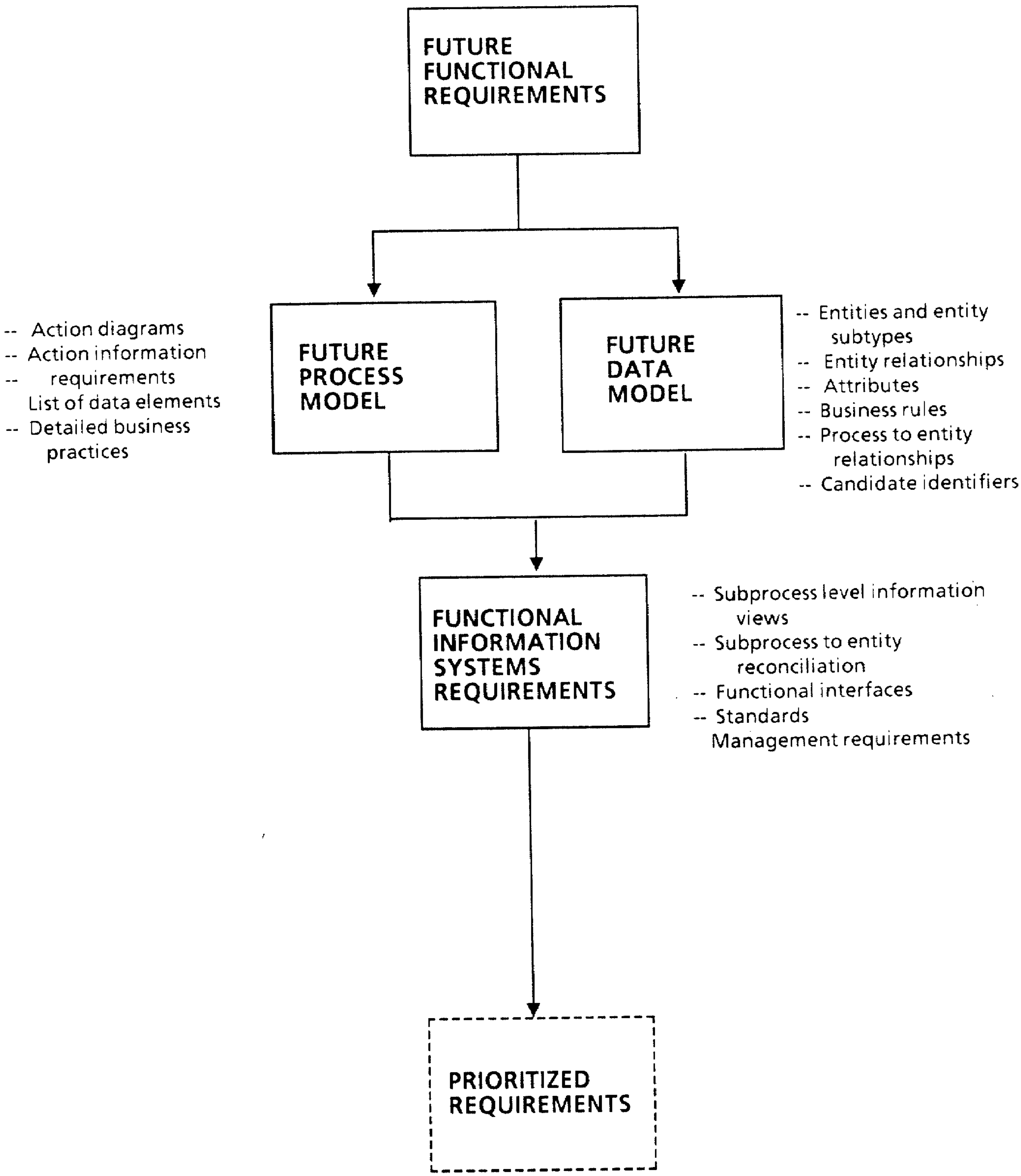
PHASE 3: INFORMATION SYSTEMS STRATEGY

- OBJECTIVES
  - TO PREPARE INFORMATION SYSTEMS FUNCTIONAL REQUIREMENTS FOR SYSTEMS DESIGN
  - TO DEVELOP A TRANSITION STRATEGY FOR MIGRATING FROM CURRENT TO FUTURE SYSTEMS
- MAJOR TASKS
  - DEFINE STABLE PROCESS AND DATA MODELS FOR NEW INFORMATION SYSTEMS ARCHITECTURE
  - DEVELOP AND PRIORITIZE REQUIREMENTS
  - DEVELOP TRANSITION STRATEGY FROM CURRENT TO FUTURE PROCESS AND DATA MANAGEMENT
  - DEVELOP REALISTIC IMPLEMENTATION STRATEGY BASED ON BEST BUSINESS PRACTICE
- TOOLS AND TECHNIQUES
  - DATA AND PROCESS MODELING
  - PRIORITIZATION BASED ON TECHNICAL, OPERATIONAL, MANAGERIAL AND POLITICAL FACTORS
  - BUSINESS PLANNING (PROJECT SCHEDULING, BENEFIT ANALYSIS, DEPENDENCE ANALYSIS)

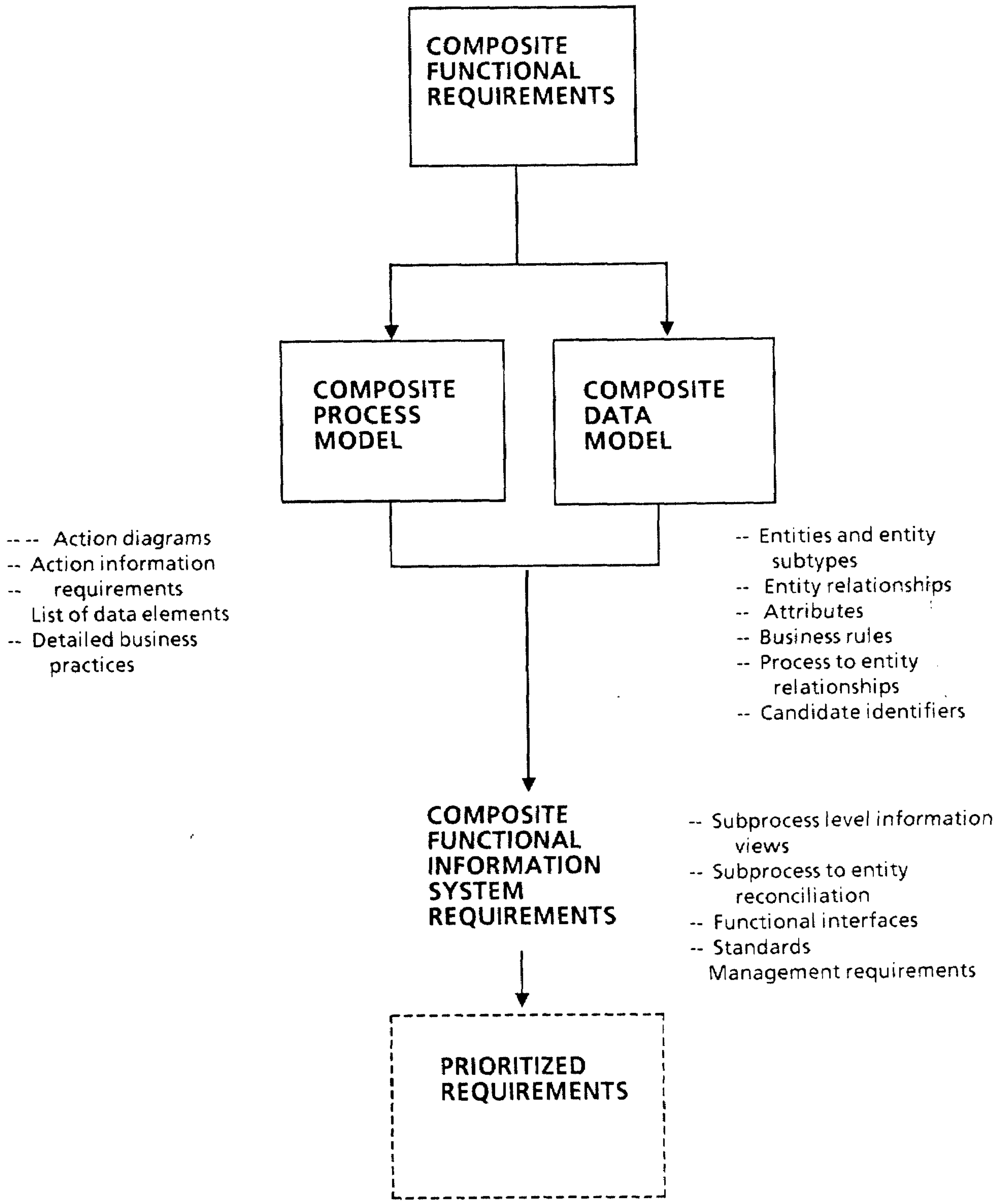
PHASE 3: INFORMATION SYSTEMS STRATEGY (CONT)

- MAJOR OUTPUTS
  - INFORMATION SYSTEMS FUNCTIONAL REQUIREMENTS
  - IMPLEMENTATION STRATEGY FOR NEXT PHASES OF SYSTEM LIFE CYCLE
  - LINK TO OTHER REQUIRED BUSINESS ANALYSES AND SUPPORT DOCUMENTS
- RELATIONSHIP TO OTHER PHASES AND TASKS
  - TRANSITION FROM STRATEGIC BUSINESS PLANNING TO INFORMATION SYSTEMS ANALYSIS
  - IMPLEMENTATION STRATEGY MARKS TRANSITION TO DESIGN PHASE OF LIFE CYCLE2

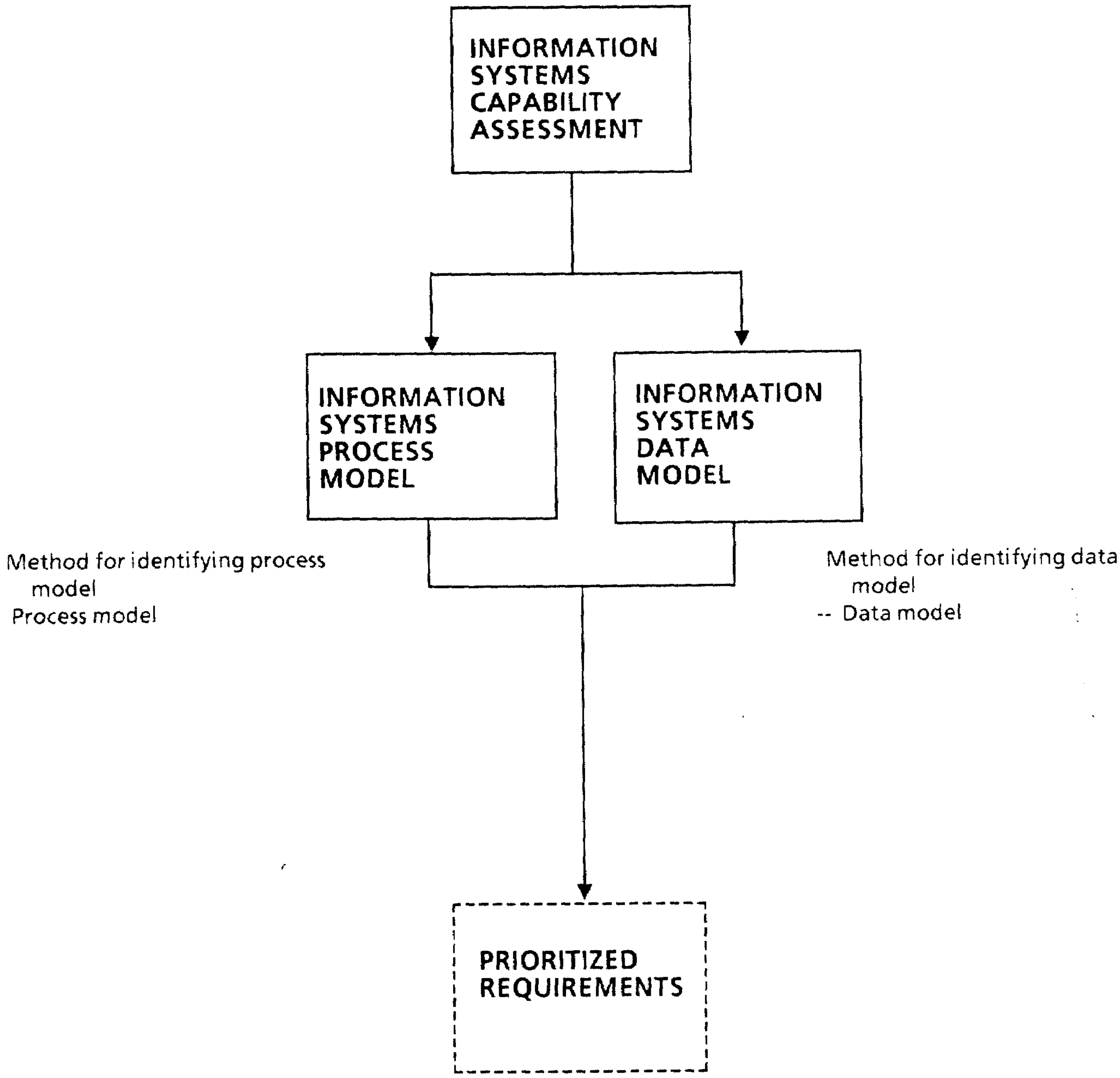
PHASE III - INFORMATION SYSTEMS STRATEGY
(Future Path)



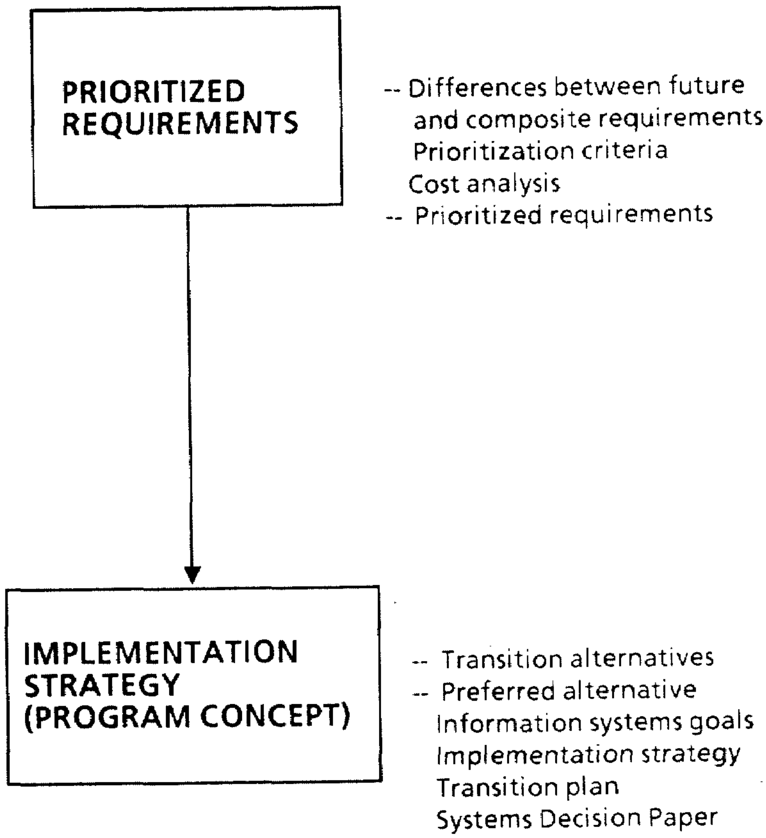
PHASE III - INFORMATION SYSTEMS STRATEGY
(Current Path)



PHASE III - INFORMATION SYSTEMS STRATEGY
(Information Systems Path)



PHASE III - INFORMATION SYSTEMS STRATEGY



AUTOMATED SUPPORT FOR PROCESS

- NEED COMPUTER AIDED SOFTWARE ENGINEERING TOOL FOR DOCUMENTATION, VERIFICATION, AND MAINTENANCE OF ANALYSIS
  - DATA MODELS
  - PROCESS MODELS
  - RECONCILIATION OF MODELS
- OTHER TOOLS SUPPORTING THE PROCESS
  - PROJECT MANAGEMENT
  - DECISION SUPPORT

ENTERPRISE MODEL AND INTEGRATION

- ENTERPRISE MODEL/INFORMATION ARCHITECTURE
- FUNCTIONAL INTEGRATION
- DATA INTEGRATION

ENTERPRISE MODEL

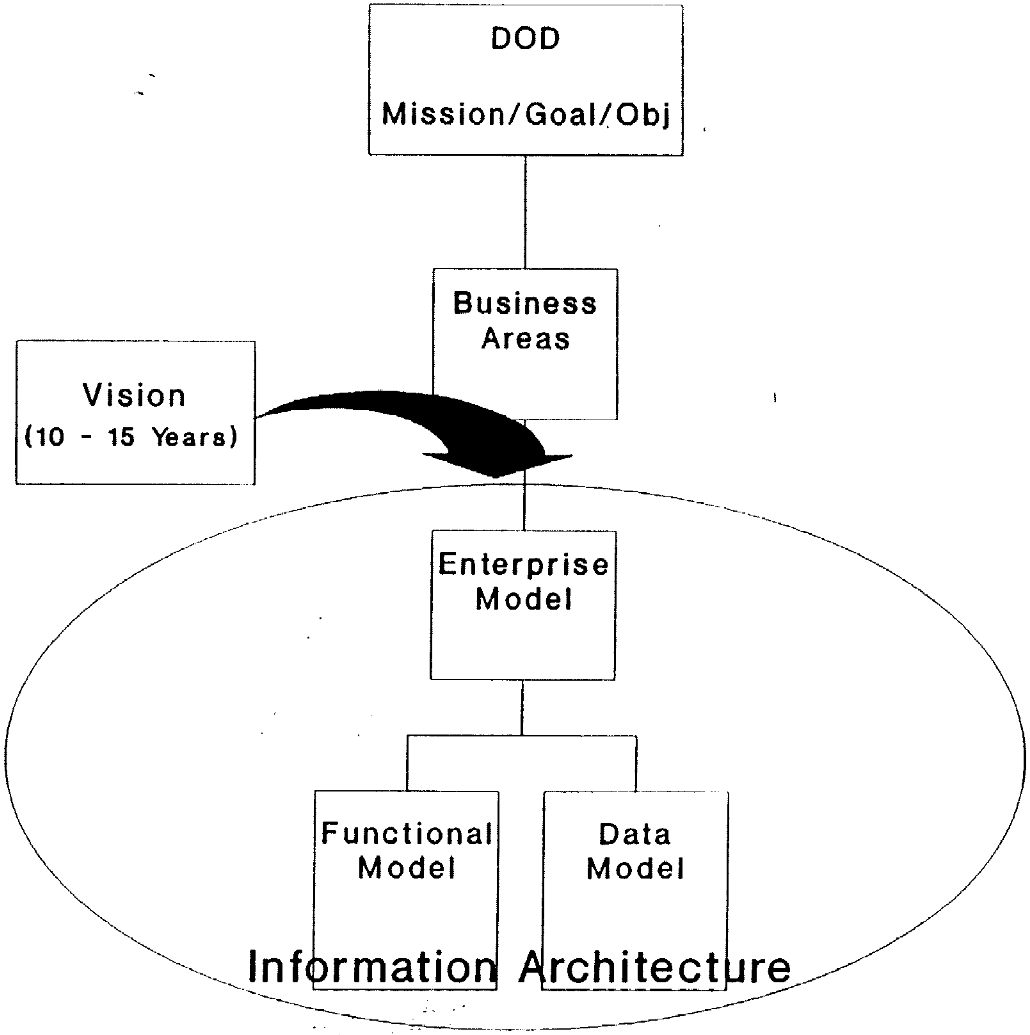
- HIGH LEVEL MODEL OF THE DEPARTMENT'S INFORMATION ARCHITECTURE
  - BUSINESS AREA/FUNCTION MODEL
  - INFORMATION/DATA MODEL
- DEVELOPED USING SENIOR MANAGERS' GUIDANCE
- ESTABLISHES FRAMEWORK FOR FIM INTEGRATION

THE ROLE OF THE ENTERPRISE MODEL

- A TOOL FOR INTEGRATING CIM FUNCTIONAL AREAS

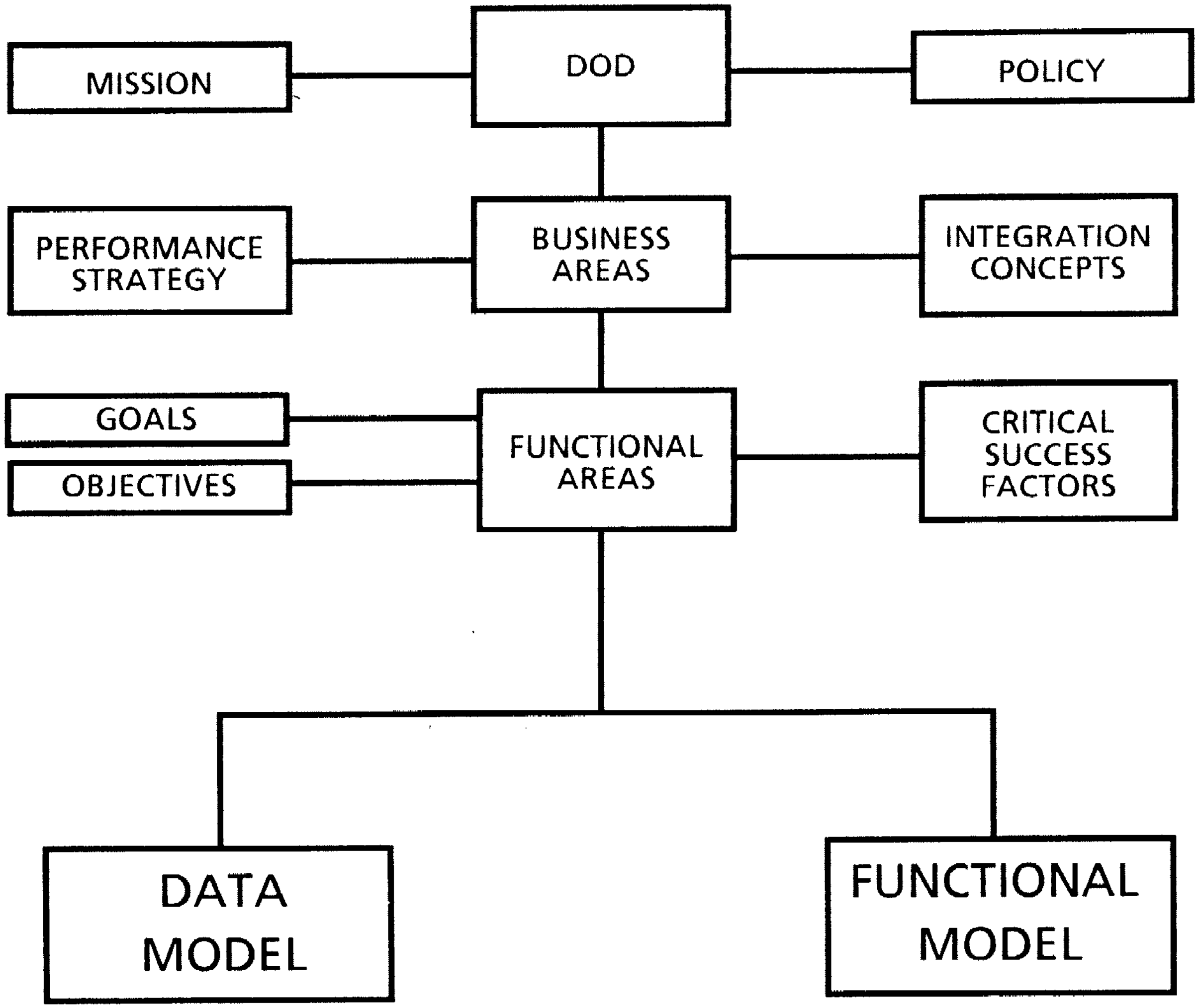
- DEVELOPED TO DEFINE BOUNDARIES AND IDENTIFY LINKAGES AND CONNECTIONS BETWEEN FUNCTIONAL AREAS

INFORMATION ARCHITECTURE COMPONENTS

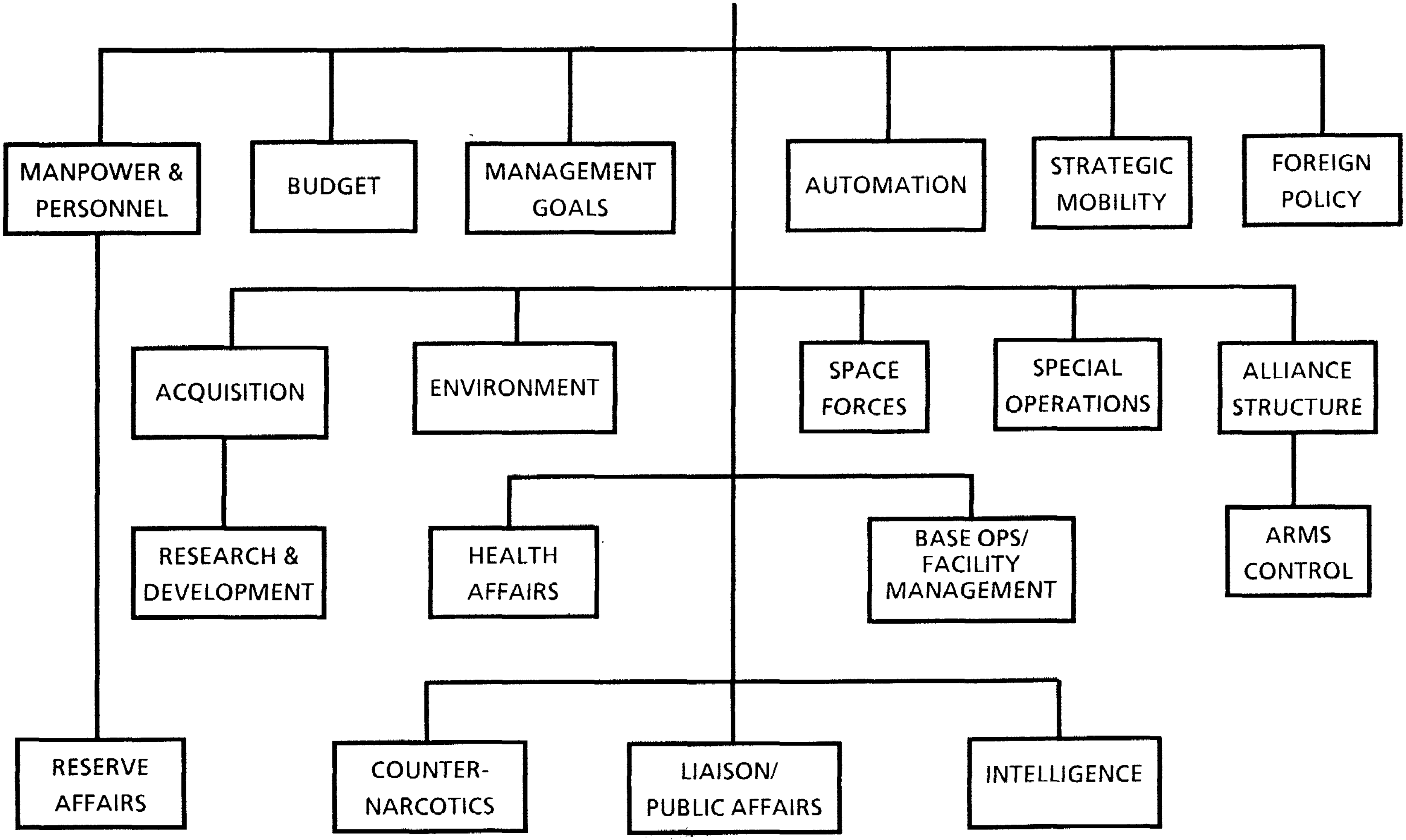


ALLHANDS-7

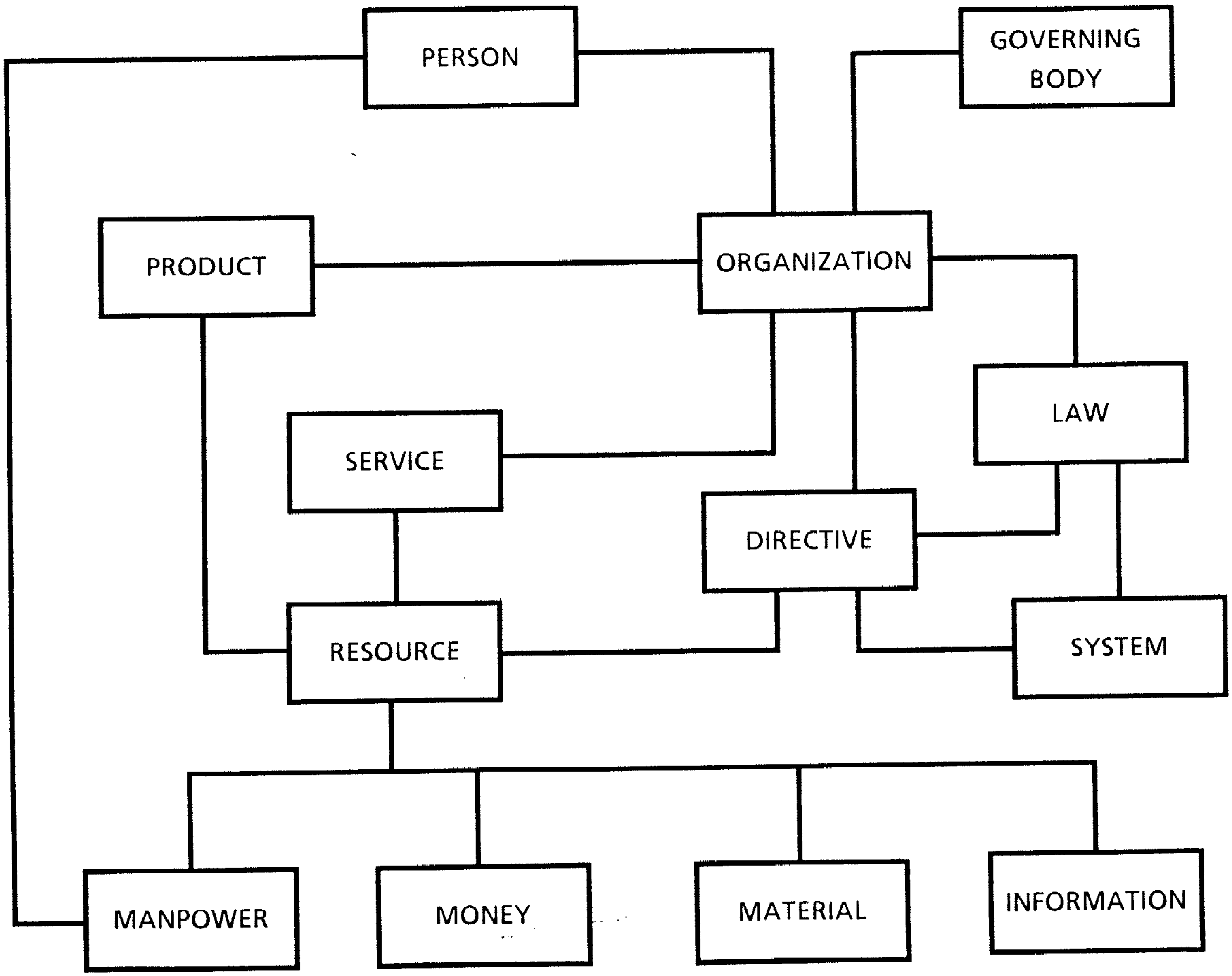
DoD ENTERPRISE MODEL



FUNCTIONAL MODEL



DATA MODEL


PERSON
GOVERNING BODY
PRODUCT
ORGANIZATION
LAW
SERVICE
DIRECTIVE
RESOURCE
SYSTEM
MANPOWER
MONEY
MATERIAL
INFORMATION

HOW WILL THE ENTERPRISE MODEL BE DEVELOPED?

- Combination of methodologies
- Carefully facilitated group interaction
- Advance preparation of analytical strawmen

METHODOLOGY
Dual Track

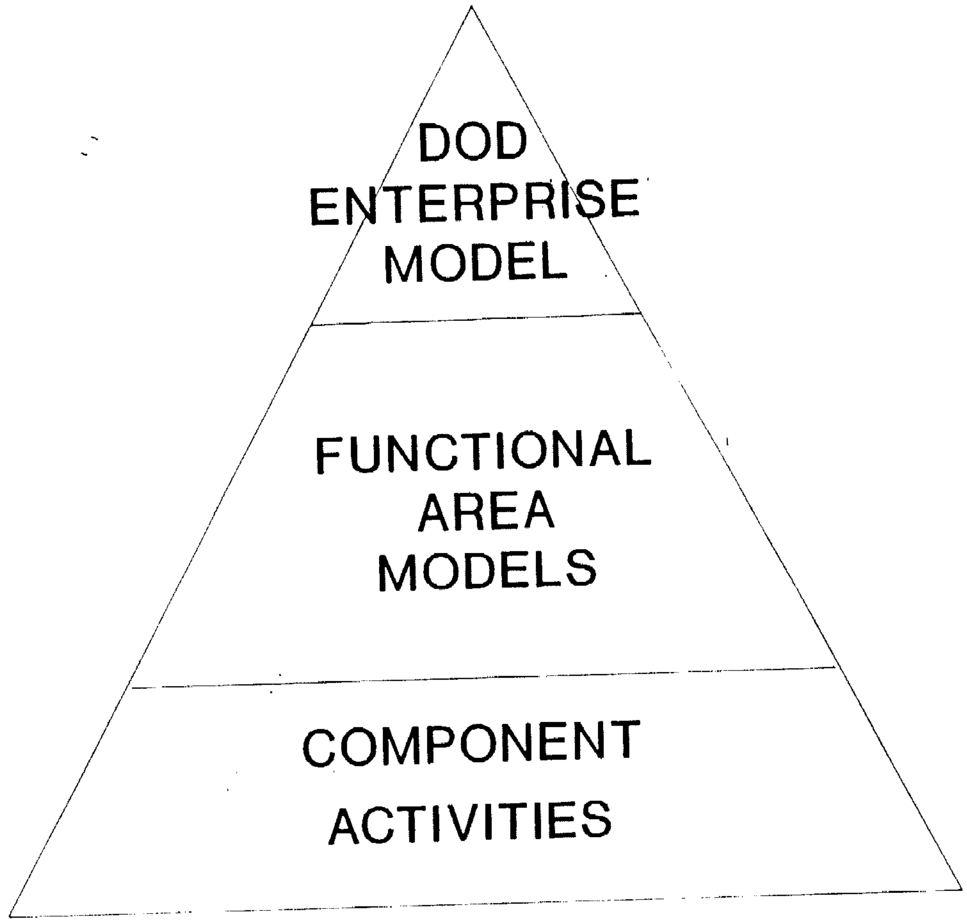
- Function/Process
  - Functional Decomposition
    - Defines *Business Processes*

- Information/Data
  - Information Engineering
    - Identifies Data and Relationships based on *Business Policy*

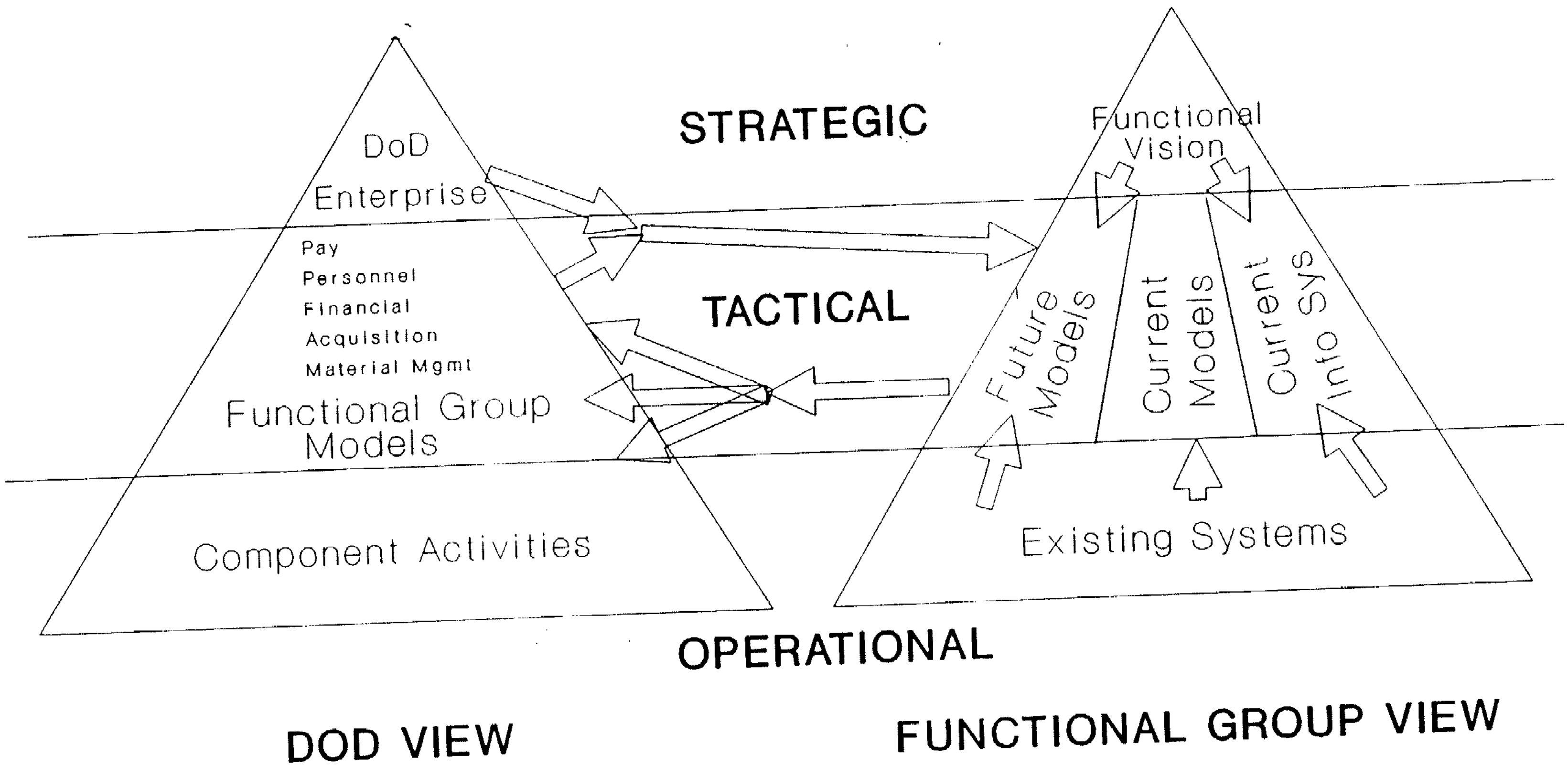
INFORMATION ARCHITECTURE DEVELOPMENT

- Analyze Defense Guidance
- Refine Vision Input from Top Management
- Develop Business Model
  Functional Model
  Data Model
- Use Standardization Rules

MODEL INTEGRATION



MODEL INTEGRATION



ALLHANDS-24

FUNCTIONAL INTEGRATION

- MAINTAINS PROCESS
  - CONFIGURATION CONTROL
  - PERIODIC REVIEW AND REFINEMENT BASED ON LESSONS LEARNED
- SUPPORTS IMPROVED BUSINESS PRACTICES ACROSS BUSINESS AREA/FUNCTION BOUNDARIES
- TIES ARCHITECTURE AND FUNCTIONAL REQUIREMENTS DEVELOPMENT TOGETHER
- QUALITY ASSURANCE OF PRODUCTS
- ASSIST IN ANALYSIS PRIOR TO DESIGN

DATA INTEGRATION

- FRAMEWORK FOR DATA MANAGEMENT BY THE FUNCTIONAL GROUPS
- ELEMENT OF DEPARTMENT-WIDE DATA STANDARDIZATION PROGRAM
- EVOLUTIONARY IMPLEMENTATION IN PARALLEL WITH STANDARD SYSTEMS DEVELOPMENT

FIM PROCESS MANAGEMENT

FUNCTIONAL GROUP LEVEL

- FIM/FACILITATORS PERFORM DAY-TO-DAY COMPLIANCE REVIEWS
- FIM REPS PROPOSE PROCESS CHANGES

INTEGRATION GROUP LEVEL

- MAINTAIN THE PROCESS
- PRODUCT QUALITY ASSURANCE
- ENSURE ARCHITECTURE COMPLIANCE

FIM PROCESS MANAGEMENT

FIM, DIRECTOR

- OVERALL PROCESS REVIEWS
- IN PROCESS REVIEWS FOR COMPLIANCE
- ENSURES PROCESS COMPLIES WITH DOD POLICY

DC (IRM)

- OVERSIGHT OF PROCESS WITHIN BROADER DOD POLICY

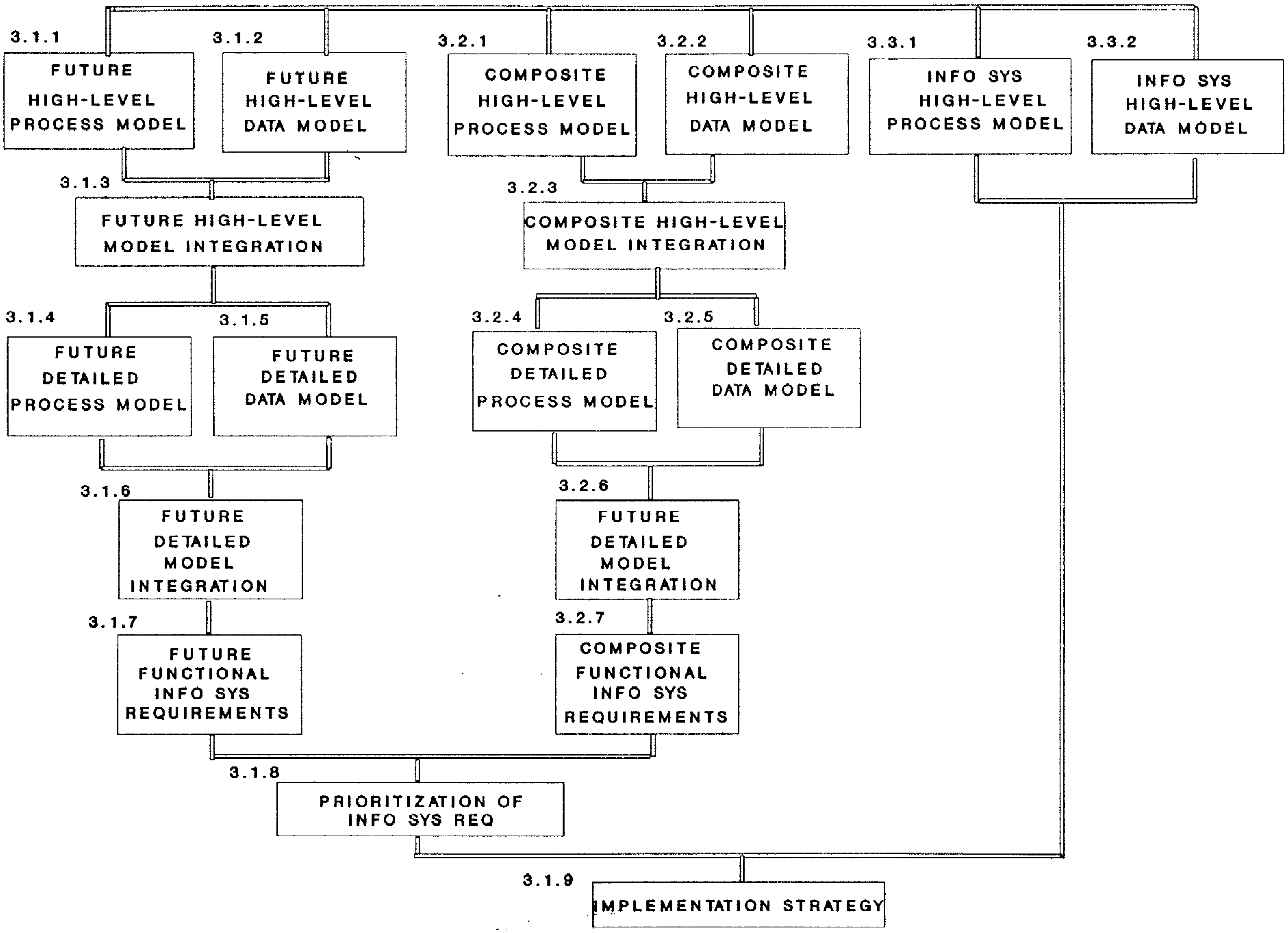
SUMMARY WRAP UP: LESSONS LEARNED

- CATALYST FOR CHANGE -- FUNCTIONAL RECOGNITION THAT IM IS MUCH MORE THAN IT
- INFRASTRUCTURE
- COORDINATION WITH KEY INFLUENCERS -- INTERNAL AND EXTERNAL
  - -POLICY DEVELOPMENT (e.g., data standards)
  - -COMPONENTS
  - -PUBLIC RELATIONS (GAO, press)
- QUALITY ASSURANCE AND CONFIGURATION MANAGEMENT
  - -PROCESS GUIDE
  - -OUTPUTS
  - -SUPPORT TOOLS

PLANS FOR FUTURE

- MANAGEMENT
  - -SUSTAINING MANAGEMENT STRUCTURE
- OPERATIONS
  - -ENTERPRISE MODEL AND INTEGRATION
  - -PROCESS STRENGTHENING
- RELATIONSHIPS
  - -LIAISON WITH COMPONENTS
- CONSTRAINTS THAT STILL EXIST
  - -STAFFING LEVELS
  - -FACILITIES

PHASE III: INFORMATION SYSTEMS STRATEGY



INTRODUCTION TO PHASE III

The Functional Business Plan from Step 2.1.9 is the foundation for the Phase III analyses culminating in a set of functional information system requirements and an implementation strategy for the information systems designers. The Phase III steps shift the analysis from strategic business planning to more specific information systems analysis. Frequent reference to the Functional Business Plan completed in Step 2.1.9 will help to assure that the Phase I and II analyses, including intermediate outputs, and the functional steering group feedback will be incorporated in the Phase III analysis. Critical products of Phases I and II used in the Phase III analysis are: the functional flow diagrams; comparative analysis of the future and composite functional requirements; issues, problems, deficiencies, and opportunities; and the action plan.

As the decomposition process proceeds, the diagramming and presentation techniques for representing the analysis takes on a new significance.

The data modeling techniques used in Phase II will continue to be used in Phase III. In Phase III entity relationship diagrams will be expanded and fully attributed.

On the process side a change is made from the use of functional flow diagrams in Phase II to the use of action diagrams in Phase III.

Phase II used functional flow diagrams to show the high-level processes in the Business Functions. This diagramming technique is well established, user-friendly, and comprehensive enough to represent the high-level data to process relationships. The effectiveness of Phase III products is measured by a different criterion: the ability to communicate the process and data requirements in sufficient detail and with sufficient clarity to permit systems professionals to design information systems without questioning the requirements or making assumptions about what was being communicated. In order to meet this objective, the detailed logical flow of the processes must be depicted with particular attention paid to sequence and process flow control.

Action diagramming technology will be used to transition the higher-level process descriptions into detailed action level descriptions. This is necessary to complete the decomposition process and to provide a sound foundation for the design phase.

At the highest level, action diagrams and functional flow diagrams are basically equivalent. They both depict the process and data requirements of the Business Function. Their differences begin to become apparent as the logic elements are defined. Action diagrams are particularly good at depicting detailed and complex process logic in a comprehensible format. As the transition from the analysis to the design activity occurs, the action diagrams remain useful and in fact can be applied all the way down to the program code level.

Although ease of use, clarity, and applicability are adequate justification for using action diagrams, another consideration is their direct automated support. Action diagrams are commonly produced via computers which eases the tedium of manual revisions. At a much later stage, action diagrams incorporating program pseudo-code can actually be used to generate the original system software. For these reasons, Phase III will rely on action diagrams to complete the process decomposition analysis.

STEP 3.1.1 FUTURE HIGH-LEVEL PROCESS MODEL

Purpose: To define the Process Model through refinement of the processes and subprocesses identified in Step 2.1.8 into their component actions. Action diagramming, a new analysis tool to depict a process model, will allow more detailed decomposition of the Business Function and serve as a basis for converting the Business Plan into information system requirements. The initial action diagram will be progressively refined in the following tasks and subtasks.

Description: In this step the group will further decompose the previously described processes and subprocesses into their component actions. Processes (and possibly subprocesses) identified in Phase II may need even further refinement before it is appropriate to attempt identifying the detailed actions. The decomposition methodology creates a continuum of more and more detailed descriptions of the processing requirement. Different Functions will reach different levels of processing when discussing processes or subprocesses. The distinction between levels is only significant when resolved down to the action level of detail.

- Actions are the lowest level of activity within the function. They are the detailed logical constructs required to perform a function.

A training session will be required to familiarize the group with action diagramming techniques. At this level, the graphic tool developed is called a Macro-Level Action Diagram and represents an abstract view of the process aspects of the Business Function.

A one sentence description of each action will be prepared. These action definitions, as incorporated into the macro-level action diagram, provide a detailed view of the function and serve as the basis for the final level of data handling analysis performed in Task 3.4.1.2.

As the processes and subprocesses are decomposed to the action level, a more specific perspective on their data requirements will become apparent. This perspective, called an information view, will be documented and used with the data model during the reconciliation process.

- An information view is a collection of data required by a process to complete its processing activity.

The information view provides a more detailed description of the information classes as defined in Task 2.1.5.4. An information class is composed of two or more logically related information views.

Task 3.1.1.1 Create Macro-Level Action Diagram

Purpose: To define the process to action relationship and to show the information flow to the action level by taking the Phase II defined processes and subprocesses and structuring them in an action diagram format.

Outputs:

- o Revalidated functional flow diagram
- o Macro-level action diagram

Relationships: The macro-level action diagram is based on the functional flow diagram produced in Task 2.1.8.3 and provides information for Tasks 3.1.1.2 and 3.1.1.3. The conversion from flow diagramming to action diagramming is required to allow introduction of more specific process details (action and logic) and to provide a sound foundation for the system designers.

Approach: In a hierarchical manner the functional flow diagram from Task 2.1.8.3 will be converted to an action diagram using a subset of the action diagramming symbols. Before beginning the action diagram, the future functional concept developed in Task 2.1.5.1 should be reviewed to confirm that the functional flow diagram is accurate and to assure that the comprehensive view of the function is well understood prior to diagramming. The action diagram will be expanded later in Phase III to reflect input and output data handling actions. The first step will only use a subset of the diagramming symbols to prepare the high-level view of the function. This diagram will be further enhanced as the decomposition process proceeds.

The action diagramming methodology is comprised of approximately 20 graphic symbols. During this task the high-level processing control symbols will be used to convert the functional flow diagram to a high-level action diagram. Figure 1, below, graphically depicts the symbols used in the action diagramming methodology and should be referenced when reading the symbol descriptions. The following narrative expands upon the graphic representations in Figure 1.

Processing Control Symbols	Description
Title	Title is the name of the function, functional activity, process, or subprocess. (See item 1 in Figure 1.) It is a processing section identifier and can serve as a qualifier for duplicative sections. Titles always begin with an asterisk. This serves to highlight and differentiate titles from the other diagramming symbols.

Titles are used in high-level discussions of the functional activities, processes, and subprocesses involved in a function. A title is usually followed by a series of actions and logic elements which specify the activities within the functional activities, process, or subprocess.

Process Rectangle

This four sided figure has several key components for defining the function processing. (See item 2 in Figure 1.) It begins with and contains Title(s). In a top-down sequence, it contains the name of the processing steps.

Outside of the rectangle <u>on the top right are the process inputs</u> (data that the process requires) and outside <u>on the bottom right are the process outputs</u> (data that the process produces). The inputs and outputs are information views, reports, products, and interprocess communications.

This identification of inputs and outputs allows for verification that every functional activity input goes into some process and that every output comes from a process. In this way, the high-level data requirements identified for each function can be traced to a decomposed process, thereby assuring that a complete decomposition has been achieved.

Initially the action diagram will be simply a graphic containing a series of Titles and Process Rectangles identifying input and output to the processes. This simple block structure must directly relate to the Functional Flow Diagram from Step 2.1.8. Through the addition of the following diagramming symbols, this fundamental action diagram will be successively decomposed into a detailed representation of the processing activities within the function.

Brackets

Brackets are used to distinguish a series of processing steps independent of input or output considerations. (See item 3 in Figure 1.) Typically they will be used within a process rectangle to highlight a series of steps which are associated in some logical fashion and are at the same level of detail.

Brackets act similarly to Process Rectangles. In fact, Brackets can be considered a short-form of Process Rectangles. They differ in that Brackets emphasize a set of actions and denote that the actions are logically related in accomplishing the processing requirement being decomposed.

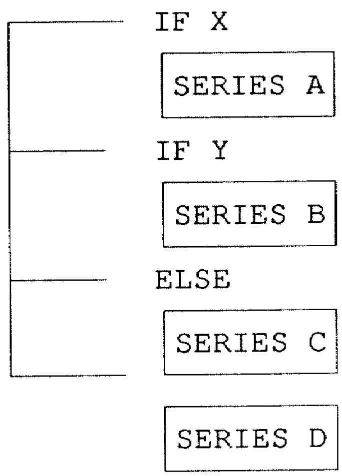
When Brackets are located inside of other Brackets, it is referred to as Nesting. This is a frequent occurrence as the Action Diagram proceeds to more detailed levels of decomposition.

IF

The IF symbol represents conditional performance of a sequence of activities. The IF symbol can appear at the top of a Process Rectangle and portray the conditions under which the processing activities occur. Typically IF symbol will be used with brackets to identify a series of processing steps subject to the same logical condition. IF symbols can contain an ELSE clause which represents activity execution when none of the IF conditions have been met.

IF symbols allow the representation of the conditions under which the following series of processing actions should be performed. That is, IF this is true, then do the following. The optional ELSE clause provides for a default processing sequence if none of the IF conditions are met.

In the example presented below, IF X is true then perform action series A. Dropping down a level, IF X is false and IF Y is true, then perform series B. Dropping one further level, IF X is false and Y is false--then perform series C identified under ELSE. After performing the "true" processing series, control flows to the bottom of the IF symbol, exits, and performs the next sequential action(s) (series D in the example).



When combined with the ELSE feature, IF symbols have a true and a false section. When the IF condition is true, only the statements following the IF condition are performed. When the IF condition is false, only the statements following the ELSE feature (if it is present) are performed. This logic can get fairly tricky if compound IF conditions (connected by OR or AND) are used. For example, "IF X AND IF Y OR IF Z" can be confusing as a condition. To alleviate some of this confusion, use simple IF symbols whenever possible. Compound IFs can often be broken down into a series of simple IFs. As a last resort, parenthesis can be used to try and clarify the meaning of a tricky compound IF. Thus, the original example could be restated with parenthesis as "(IF X AND IF Y) OR IF Z".

Nesting

Nesting represents a processing activity which contains one or more lower level processing activities. (See item 6 in Figure 1.) It is depicted by a smaller Process Rectangle completely enclosed within a larger Process Rectangle. All of the basic rectangle features apply to a Nested Rectangle. As noted previously, Nesting will most often occur at the Bracket level.

When Nesting symbols are used in combination with IF symbols, the results can get a little tricky. Particular care should be taken when combining IF symbols and Nesting to assure that the process flow described in the diagram is the one required by the function.

Repetition

Repetition is the process of repeating an activity or a section of activities several times. (See item 4 in Figure 1.) This is represented by a double bar at the <u>beginning</u> and <u>end</u> of the repetition sequence. Repetition control structures or qualifiers can be identified at the top of the sequence, much like a Title. Such qualifiers can include DO WHILE, LOOP WHILE, FOR, REPEAT, and others. Since the analysis is language independent, all that is important is consistency in applying the repetition nomenclature.

If Repetition is viewed simply as a way to avoid having to duplicate a processing description, its use remains fairly straightforward. However, instances of Repetition, particularly when combined with IF/ELSE symbols and Nesting can significantly increase the complexity of the process descriptions. As the objective of Phase III is analysis and not design, use of Repetition should be logically driven by the process requirements and not used as a way of representing "how" the processing should be performed.

EXIT

The EXIT symbol represents termination of the activity sequence and rescinding control to another activity sequence. (See item 5 in Figure 1.) EXIT can be used as an end of processing placeholder (the bottom) whereby control is returned to the next higher level of processing. Alternatively, EXIT with an arrow attached can represent an immediate branching from this processing level to another processing level. In this role it could be considered a form of GOTO. EXITs can be conditioned by using IF or Repetition control structures, as in IF X=Y EXIT.

An EXIT is simply a form of escape which allows redirecting the flow of processing. Often it is used to end a repetitive series of processing activities to return to a higher level of processing. Although the EXIT symbol is an easy way to jump out of a Nested series of Brackets or Process Rectangles, it seriously diverts the flow of processing from a sequential top-down flow. For this reason, EXITs should be used sparingly and only when they simplify the logic. When an EXIT seems necessary, examine the IF symbol conditions and see if they can be modified to eliminate the need for an abrupt EXIT.

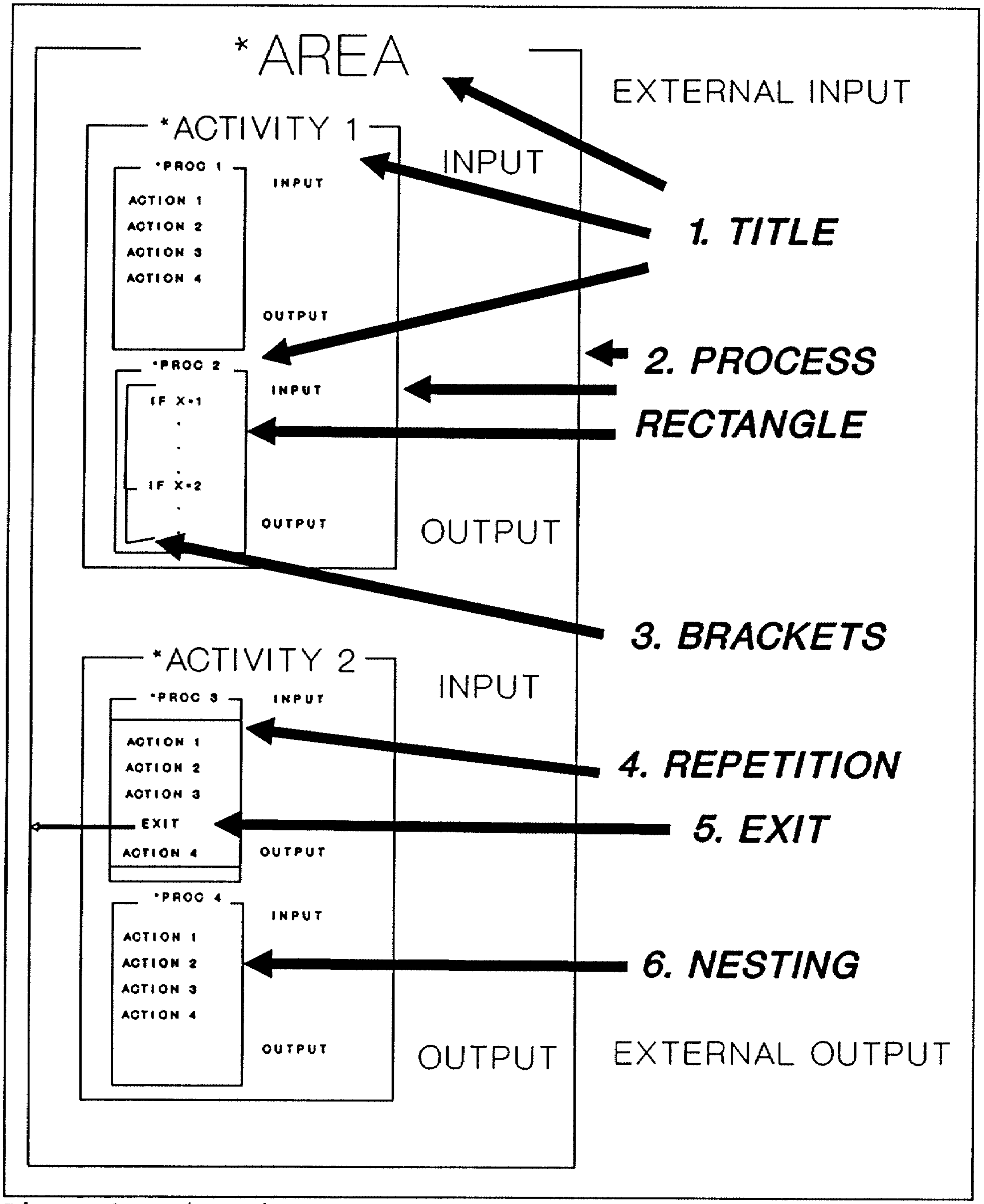


Figure 1 Action Diagram

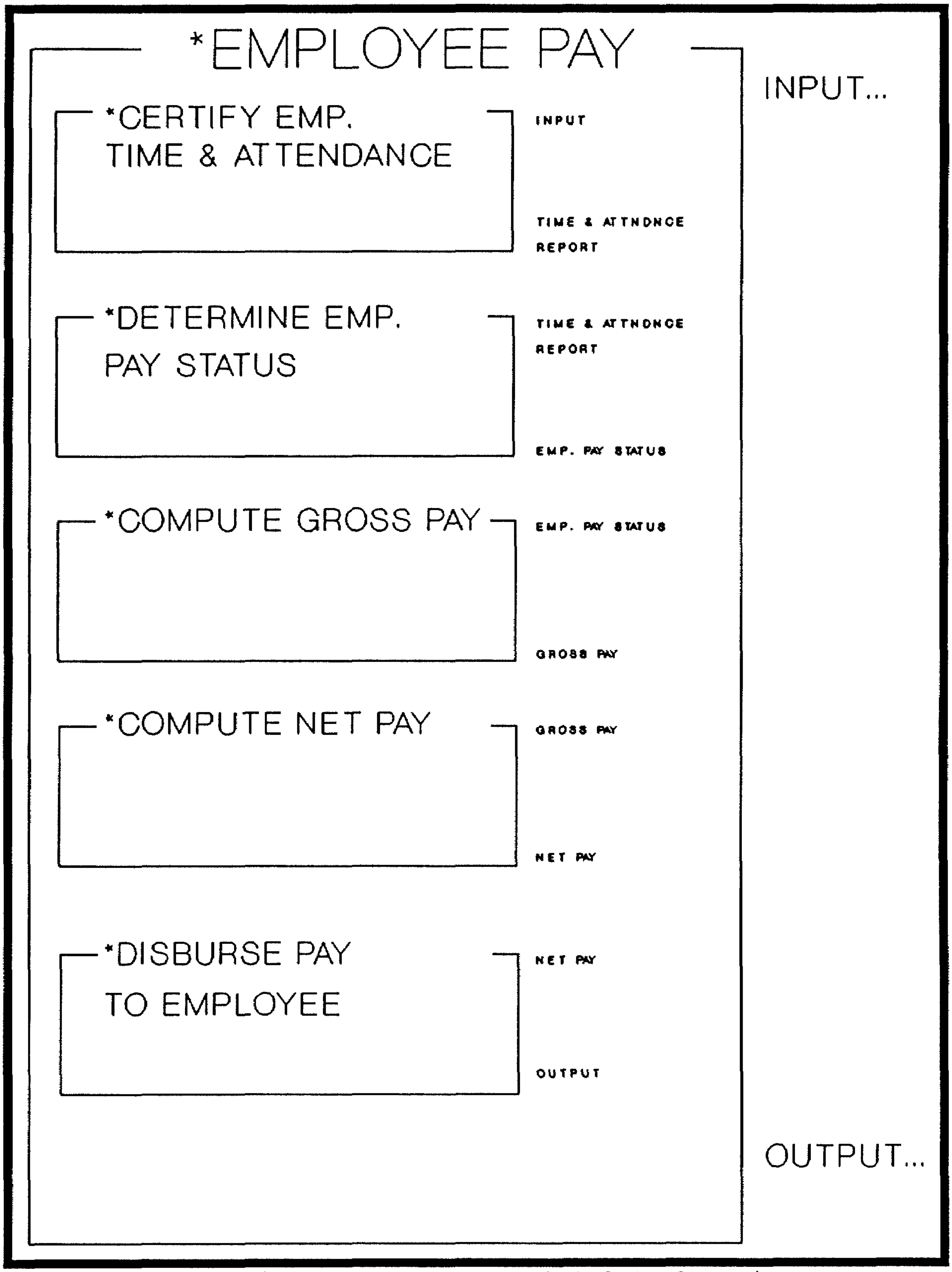


Figure 2 Action Diagram of the Step 2.1.8.1 Flow Diagram

Subtask: Review Future Functional Concept and Confirm the Accuracy of the Functional Flow Diagram

The functional flow diagram produced in Task 2.1.8.3 should be reviewed to verify that the Function is clearly and accurately represented in the diagram. The functional flow diagram must be intensively reviewed to assure accuracy, as it provides the "big picture" view of the business area and serves as the point of departure for the detailed process decomposition. Processing bubbles identified in the flow diagram should accurately represent the Function and be established in a fashion which will permit further decomposition. Within a function, processing area overlap and duplicative data handling responsibilities should be considered for elimination. Major data flows should be identified reflecting their associated (source and destination) process bubbles and relationships. The output of this subtask should be a final version of the functional flow diagram which rigorously presents the processing and data used in the Function.

Subtask: Convert Processes and Subprocesses to Actions and Identify Input and Output (Information Views).

This subtask is begun by preparing a large processing rectangle to contain all of the functional activities, processes and subprocesses defined previously in Phase II. This rectangle should be titled with the name of the function being analyzed. It is critical that all external inputs and outputs to the function be listed in the appropriate locations outside of this box. These external information views represent this function's linkage with other functions.

- o If information used within a function is obtained or given to a process outside of this function, then that information view must be listed in the appropriate location (<u>input at the top right, output at the bottom right</u>) outside of the largest functional processing rectangle. These entries represent this function's external interfaces.

Through direct reference to the functional flow diagram produced in Task 2.1.8.3, process rectangles should be constructed which reflect the sequential and hierarchical structure used in the flow diagram.

Process and subprocess input and output requirements should be listed in the appropriate areas of the action diagram in a manner which reflects a "trickle down/float up" structure. This should make apparent where inputs and outputs to the processes and subprocesses originate and their flow from process to process should be evident. There is no distinction in the diagram between process to process communication and process to data store to process communication.

Figure 2 illustrates the format and level of detail required for this subtask. This example is an action diagram of the flow diagram presented in Task 2.1.8.1. The Employee Pay model contains the series of processes and subprocesses identified in the processing rectangles. These processing rectangles will be further developed in the next subtask to identify the actions required to accomplish the processing. In the example not all of the input and output requirements were provided. However, input and output information views are integral to the decomposition process and should be analyzed with an intensity equal to that of the process decomposition.

The information views (and their data subcomponents) are critical for comparison with the data model during later steps in Phase III. Note that for external data, only the information views (and data sources) are documented, not the processes that produce the data. This source and destination documentation is primarily used as an aid to the design phase and to identify opportunities for integration across functions.

Subtask: Define Processing Control Structures

The conditional control logic required by some of the processes is identified in this step. The addition of process control structures clarifies the order of processing and identifies any conditions which influence the order of processing. Principally this involves the identification of the actions which comprise a process (what processing activities need to be performed) and the definition of IF structures and IF/ELSE conditions.

Potentially these structures can cause the repetition of some processes or subprocesses in order to clarify the control logic. This duplication need not be avoided. However, once the IF control structures are defined, the overall diagram should be reviewed to determine if it can be simplified by reordering or restructuring some of the processing activity sequences. Complex IF symbols should be reworked to simplify them as much as possible. This may require breaking them into multiple IF structures. This restructuring should only make use of Nesting and EXIT symbols to represent the processing logic when their use simplifies or clarifies the flow at processing.

As the decomposition process proceeds to the action level of detail, it will be apparent that the action diagram has expanded beyond a single page. This is the transition from a macro-level action diagram to a standard action diagram.

If an automated symbol is being used, there should be no difficulty in taking a modular approach to the diagramming activity. The initial process level of detail (as performed in the previous subtask) should be retained on as few physical and logical pages as possible. This will help in gaining an overall perspective of the function much as was attained in the original functional flow diagram. However, as the processes and subprocesses are decomposed into actions and control logic, they will probably require the use of more than one page.

Consistency across diagrams and pages regarding control structures and nesting levels is critical. Due to the number of details and actions which need to be diagrammed, as well as the normal iterations occurring in any analytical activity, an automated diagramming symbol is recommended.

Subtask: Cross-Function Integration

It is not unusual for a functional group to find they have identified processing requirements which should be the responsibility of another function. For example, the Contract Payment group (or possibly the Materiel Management group) might identify a processing requirement to maintain a vendor or contractor list. In fact, this process is probably best performed by the Contract Management group who intuitively would have primary responsibility for the list of contractors. To avoid redundancy and confusion, these duplicative areas should be coordinated with the affected functional groups and a determination made as to assigned responsibilities (many users, one owner). In coordination with the other functional groups, a structured walk-through of the action diagram should be presented. This will allow the group to review its diagram for completeness and quality of concept. A briefing should be prepared as a discussion vehicle, and in coordination with the other functional group, a meeting should be held to identify and delineate areas of responsibility.

Task 3.1.1.2 Develop Future Actions Definitions

Purpose: To further refine the action level process decomposition by defining all of the identified actions.

Outputs:

- List of action definitions

Relationships: The list of actions is obtained from the macro-level action diagram created in Task 3.1.1.1 and is used in the reconciliation process in Step 3.1.3.

Approach: Obtain a list of action names from the macro-level action diagram and explain their meaning using one sentence narratives. Typically this will include all of the lines which do not have either a title or processing control symbol.

The action names should be qualified by process or subprocess for further clarification and a list prepared compatible with the existing process definitions (see Task 2.1.6.1). Higher order actions, which are groups of more primitive actions, should be broken down to the lowest action level prior to definition. Actions which cannot be clearly defined in a one sentence definition should be considered for further decomposition. It is not unusual for higher order actions to masquerade as primitive actions to this point in the analysis. Through the diagramming and definition processes the work group should resolve the actions to their primitive level to complete this step.

An action can be considered adequately resolved when its purpose can be described in one sentence. For example, in the Contract Payment Function "authorize payment of a contract" could be an action definition. The action and definition should address the "what" aspect of the function and must not be influenced by the "how" design aspect.

Task 3.1.1.3 Define Compoiste Action Information Requirements

Purpose: To identify the information requirements (data elements) of the function at the action level. Detailed data specifications which have been casually identified during the decomposition process will be collected and defined for later use during the reconciliation with the data model.

Outputs:

- o List of information requirements (data elements) and definitions

Relationships: This list is obtained from the macro-level action diagram created in Task 3.1.1.1 and is used in the reconciliation process in Step 3.1.6.

Approach: The input and output requirements (information views) identified in the action diagram will be used as reference points for identifying some of the data element requirements of the processing actions.

- o Data elements are elementary types of data which collectively form a processing information view. Data elements relate directly to the data attributes identified in the data modeling process. For example, when viewed from a processing perspective the information view called employee may be composed of the service identifier, social security number, age, sex, and marital status data elements (among others).

This task serves two major functions. It encourages clarification of the information views required in the 3.1.3 Future High-Level Model Reconciliation step and begins the collection of data elements required for the 3.1.6 Future Detailed Model Integration step.

Typically the action diagrams identify information views at the process level and above. At the action level it is not unusual to identify major data elements as well as information views. In this task any data elements which have been recognized while identifying the information views will be organized into a list of definitions. <u>This should not be an intensive dissection of the information views, but rather the collection of incidentally or already apparent data elements.</u>

Using the input and output sections of the action diagram as a starting point, examine the actions to collect any data elements which have already been identified and to consider what data elements are required to perform the processing. For this analysis the actions should be viewed as processors which transform the input into the output. All data elements identified as needed to perform this transformation should be listed and defined in a short narrative form. This narrative should specify any apparent special roles (key, index, etc.) which this data element performs for the information view.

STEP 3.1.2 FUTURE HIGH-LEVEL DATA MODEL

Purpose: To continue describing at a high-level the data portion of the future information requirements for the function.

Description: In Phase II the Future Functional Information Model was created. In this step the group will continue to use Entity-Relationship diagramming techniques to build upon this model and to create the initial high-level Future Data Model (high-level in the sense that not all entities and entity subtypes may be identified and that it will not be fully attributed). This model will be compared with the set of action diagrams developed in Step 3.1.1 to reconcile specific information processing requirements and to maintain consistency between the data and process perspectives of the function.

In the Future Functional Information Model from Step 2.1.7 only major entities were identified. In this step the group will identify all other entities and will begin to identify entity subtypes. Candidate identifier attributes and other significant attributes identified during the data modeling process will be documented in the standard Corporate Data Dictionary. Relationships among entities and entity subtypes will be captured. An entity-relationship diagram will be drawn.

Integration of entity descriptions, entity subtype structures, and attribute definitions with other business functions and with other analytic paths (i.e. the composite functional path and the current information system paths) will occur in the final task of this step.

This step will require the assistance of a trained data modeler.

Outputs:

- High-level data model consisting of
  - Entities and descriptions
  - Entity subtypes and descriptions
  - Entity and entity subtype relationships
  - Relationship descriptions
  - Entity/entity subtype structures
  - Entity relationship diagram
  - Business rules
- Candidate identifier attributes

- Attribute names and definitions
- Updates to the Corporate Data Dictionary

Relationships: In Step 3.1.3 the data model will be reconciled with the action diagram from Step 3.1.1 to ensure that the processes required to effectively manage data entities have been identified and to determine whether the information requirements (action diagram information views) necessary for the successful execution of processes can be satisfied.

The entity descriptions, entity subtype structures, and attribute definitions identified in this step will be integrated with similar products for other business functions to promote corporate wide data integrity.

Task 3.1.2.1 Identify Entities and Entity Subtypes and Document Their Descriptions

Purpose: To identify entities and entity subtypes necessary to support the function and to develop descriptions of these entities and entity subtypes.

Outputs:

- o Entities and entity subtypes
- o Entity and entity subtype descriptions

Relationships: The entities and entity subtypes with their descriptions will be used to build the entity relationship diagram in Task 3.1.2.6.

Approach: This task will be accomplished through three subtasks.

Subtask: Identify Entities

- o A data entity is something of lasting interest which is uniquely identifiable and about which data must be stored. An entity can be tangible such as a person, place, or thing; or intangible such as an event or concept.

In this step the group is not building an entirely new data model, but rather is continuing to build on the model begun in Phase II. With the high-level entities from the future Functional Information Model in Phase II as a reference, the group will identify additional entities necessary to execute the function.

In Phase II the group was encouraged to identify only major high-level entities. In this task the group will consider all entities and entity subtypes necessary to support the function, including all entities needed to support the individual processes and subprocesses identified in the High Level Action Diagram from Step 3.1.1. To ensure that information linkages to other functions are developed in this task, both the entities that lie within the function and those that are closely related but lie outside the function must be identified. The latter type of entity is best identified by analyzing the descriptions of external interfaces defined in Task 2.1.6.4.

The same series of tests used in Task 2.1.7.1 to identify entities for the Future Functional Information Model may be applied to determine if candidates are in fact entities:

- o Is it necessary to collect information about the candidate in order to manage or execute the function? If yes, what kind of information?

- o Does it have meaningful relationships with other potential entities?

- o Is it distinguishable from other potential entities?

- o Can a single occurrence of the potential entity be uniquely identified?

Since we are representing the data model of the future the group must be open minded and creative in its identification of entities. The future vision should be closely appraised in identifying candidate entities.

The group must also be careful to consider those entities appropriate to the scope of the future function. Entities which are within the scope of the current function may no longer be within scope in the future. For example, the current financial function may manage entities which deal with people. In the future the management of "people" entities might belong entirely to personnel. "People" entities may be referenced by the future financial function, but should not be managed there. Such entities will be identified, but they will be part of the external interface of the function.

The entities identified in this task, together with those from Phase II, will become the basis for the entity-relationship diagram to be developed in Task 3.1.2.6.

Subtask: Identify Entity Subtypes

While developing the Future Functional Information Model in Phase II the group was specifically directed not to identify entity subtypes. In this step the group will identify subtypes, as appropriate. In general, this step should identify subtypes to the first level below the entity.

- o Entity subtypes are a hierarchical decomposition of an entity based on some criteria. They are subsets of an entity established to record information specific to the subset and which have distinct associations to other entities. Ideally subtypes fully partition the entity (every instance of an entity will belong to one of the subtypes) and there will be no overlap (each instance of an entity will belong to only one subtype).

For example, Individuals may be either Civilian Employees, Military Members, or Customers. It may not be sufficient to identify the single entity, Individual, and accurately model the function's data. The function may treat Civilian Employees differently than Military Members and Customers. Since each of these are categories of the entity type Individual, they are all subtypes of the Individual entity. Individual, in turn is considered a supertype of the entities Civilian Employee, Military Member, and Customer. A relationship originally expressed as "Individual places Customer Order" could be more meaningfully described as "Customer places Customer Order". This can only be done if Customer is identified as an entity subtype.

There are generally multiple ways to partition an entity into entity subtypes. Individual could be subtyped as male or female, active or retired, or by race just as easily as we partitioned by Civilian Employee, Military Member and Customer. The way the data is partitioned into subtypes changes the way the data is viewed. The method of subtyping selected should be that which most meaningfully represents the data as viewed by the business function. This will be determined by the data steward (described in Task 3.1.2.7) for the entity.

The following logic can be used to determine if an entity subtype exists:

Entity B is a subtype of entity A if

- entity B and entity A represent the same object in the real world; and
- entity B has all of the attributes (properties) of entity A plus some additional attributes of its own (This will be clarified in a later step when entities are fully attributed.); and
- for every occurrence of entity B there exists precisely one occurrence of entity A while the reverse is not necessarily true, i.e., there may not be an occurrence of B for every A.

Even where these criteria are met, entity subtypes should only be identified where they serve some clear business purpose and not merely as an academic exercise. Upon close examination virtually all entities can be categorized into subtypes. The question must be asked whether the entity subtype is treated differently (has different relationships to other entities) in the business than the entity supertype. If the subtypes are not treated differently, there is no reason to formally document them.

Conceptually, there may be multiple levels of entity subtypes. Military Members (a subtype of Individual) may be categorized as Officer and Enlisted. Officer (a subtype of Military Member) may be Active, Reserve, or Retired. It is possible to spend considerable time exploring the levels of entity subtypes and the resulting data structures.

The purpose of this step, however, is not to explore the data to its lowest level. This will be accomplished in a later step. Rather, the group is trying to establish a data model which will clearly represent the information requirements of the functions and to do so as expeditiously as possible. It is anticipated that this can be accomplished by structuring the data model to the first level of entity subtypes. If further decomposition is clearly called for to establish meaningful relationships, then it is certainly acceptable to identify those subtypes now. Additional decomposition may also be accomplished in Step 3.1.3 as the information views necessary to satisfy specific information requirements at the action level are reconciled.

A preliminary activity to the identification of entity subtypes should be the identification of any undiscovered entity supertypes. It is possible that in the original identification of entities some number of entities were identified which were actually subtypes of the same supertype. For example, we may have identified Civilian Employee and Military Member as entities without having recognized that they are both subtypes of the same supertype - Individual. While it is our intent to ultimately identify these subtypes, the supertypes must be established first if subsequent modeling is to proceed in an orderly fashion. In general, entities are of the same supertype if they have common identifying attributes and do not have clearly distinct business meanings.

Subtask: Describe Entities and Entity Subtypes

In this step the group will capture descriptive information about entities and entity subtypes, including a definition and a statement of the business purpose for each entity or entity subtype described.

Each entity must be uniquely and unambiguously defined so that potential users who are not involved in the analysis effort will be able to determine exactly what is included in, and excluded from, the entity. A precise definition of each entity is essential. Ambiguities may lead to redundant and inconsistent implementation. Definitions will focus on what the entity is and not how it is used.

Entities identified during Phase II were described at that time. The group has since acquired a more thorough understanding of the function and of its data. The Phase II descriptions will be reviewed in light of this increased understanding and modified as appropriate.

Once entities are described, entity subtype descriptions must also be developed. Each entity subtype must be described in a manner which makes clear the criteria used to distinguish it from the entity supertype and from all other entity subtypes at the same level.

Task 3.1.2.2 Identify Candidate Identifier Attributes

Purpose: To identify candidate attributes which uniquely identify each instance of an entity. To record the definitions of these attributes in the Corporate Data Dictionary.

Outputs:

- Candidate identifier Attributes
- Updates to the Corporate Data Dictionary

Relationships: In ensuing tasks the group will identify relationships among entities and will describe the characteristics of those relationships. The ability to accurately and completely identify relationships depends on the level of understanding of the entities in the model. Documentation of candidate identifiers will enhance this understanding and facilitate identification of relationships in Step 3.1.2.4.

Candidate identifiers will become part of the future information requirement.

Approach: For each entity or entity subtype one or more candidate identifiers will be documented.

- Attributes are characteristics of data entities which are describable in terms of some value. They are the lowest level of information about data. For purposes of the Phase III Process Guide, attributes are synonymous with data elements.

A candidate identifier is an attribute of an entity which uniquely identifies a given instance of the entity from all other instances of that same entity. Name, Social Security Account Number, Employee ID, Address, and Phone Number might all be attributes (or properties) of the entity Employee. In this case Social Security Account Number, Name, and Employee Id would all be candidate identifying attributes which might uniquely identify each instance of the entity Employee.

Sometimes multiple attributes are necessary to uniquely identify an entity. Both the Flight Number and the Date might be necessary to uniquely identify each occurrence of the entity Airline Flight. In this case a concatenated, or compound, identifier can be established consisting of multiple attributes.

The identifier of an entity subtype is generally the same as the identifier of the supertype. Sometimes, however, additional attributes are necessary to fully qualify instances of a subtype. In this case the additional attributes should be concatenated and the resultant compound identifier used as the identifier for the subtype. In other cases completely different candidate identifiers may exist. Customers and Employees might both be subtypes of Individual. While Social Security Account Number would seem like a strong candidate identifier for Individuals, we might not capture this information about either the subtype Employee or the subtype Customer. A candidate identifier for Customer would be Customer Account Number, while for Employee we could use the Employee Account Number.

While it is tempting to focus on a single identifier at this point, caution should be exercised. The best identifier may be different for different functions. Selection of the identifier may be deferred until the implementation of physical data structures.

As candidate identifying attributes are identified they will be named in accordance with corporate standard naming conventions documented by the corporate data administrator. Final approval of all attribute names must be by the corporate data administrator and must not be accomplished by individual group members. This control over attribute naming is key to preserving data integrity and is a first step in integrating data across functions.

Task 3.1.2.3 Document Additional Attributes

Purpose: To collect information about those additional attributes (beyond the candidate identifier attributes identified above) which the group feels are necessary to document at this point.

Outputs:

- o Attribute names and definitions
- o Updates to the Corporate Data Dictionary

Relationships: Attribute definitions will be entered in the data dictionary for reference by functional groups, central design activities, and other interested parties throughout the life cycle of the business function.

Approach: It is not the purpose of this task to fully attribute data entities. Rather, the identification of attributes beyond the candidate identifiers should be deferred until Step 3.1.5 whenever possible. This task recognizes that during the modeling process common attributes are often identified which the group may feel are significant and worthy of immediate documentation.

New attributes will be named (in accordance with the standard naming conventions mentioned above), defined, and related to the appropriate entity or entity subtype. Entries for all new attributes will be made in the Corporate Data Dictionary under the control of the corporate data administrator.

As attributes are identified, the existing Corporate Data Dictionary will be researched to determine if the attribute already exists and if the definition as written is applicable. Problems with existing data names and definitions are data administration issues which must be resolved under the authority of the corporate data administrator and consistent with the concept of data ownership discussed in Step 3.1.2.7.

The principle of inheritance will be applied in the identification of attributes.

- o The principle of inheritance allows lower level subtypes within a hierarchy to inherit attributes from the higher level. An attribute will be included at a higher level only if it applies to all of the subtypes. An attribute which only applies to some of the subtypes will be described at the level of the subtype. For example, all mammals are warm blooded. The attribute "warm blooded" would apply at the level of mammals. Only some mammals walk on four legs. The attribute "walks on four legs" would not apply at the level of mammal, but rather would apply individually to each of the mammal subtypes which walk on four legs (giraffes and horses, but not apes and man).

In accordance with the principle of inheritance, attributes will be related to the highest appropriate level within an entity/entity subtype hierarchy. Consider our categorization of the entity Individual into the entity subtypes Civilian Employee, Military Member, and Customer. Since the attribute Name applies to all subtypes (Civilian Employee, Military Member, and Customer) of the supertype Individual, the attribute should be related to Individual and will be "inherited" by the subtypes. It therefore becomes unnecessary to relate the Name attribute to each of the subtypes individually. The attribute describing "customer account balance" would meaningfully apply only to Customer and would therefore be captured as an attribute at the level of the subtype, Customer.

The amount of attribute information collected at this point may be minimized. Collection of detailed information about default values, optionality, security, and the like may be deferred until Step 3.1.5.

TASK 3.1.2.4 Identify Relationships Among Entities and Entity Subtypes

Purpose: To identify relationships among entities and entity subtypes.

Outputs:

- o Entity and entity subtype relationships
- o Relationship descriptions
- o Entity/entity subtype structures

Relationships: Entity relationships will be used to build the entity relationship diagrams (ERDs) in Task 3.1.2.6.

Approach: A relationship is an association among entities or entity subtypes. Relationships have properties that describe what they are and how they operate.

Relationships among entities and entity subtypes are expressed in the form of a verb or verb phrase. The relationship between the entity Students and the entity Classes is that Students "attend" Classes and Classes "are attended by" Students. This is the most common type of relationship on an ERD and is the type that was introduced in Step 2.1.7.

There is, however, a second type of entity relationship. This is the relationship that exists between levels within an entity/entity subtype hierarchy. That is, the relationship between a high level entity (entity supertype) and what may be called its composite entities (entity subtypes).

The distinction between an entity and its entity subtypes exists as a result of the categorization of an entity into subtypes based on some property or attribute. This concept works much the same as the way animals are categorized based on whether they are cold or warm blooded, whether or not they eat meat, or whether they have two, four, or more than four legs. In each case a single attribute may be used to categorize into subtypes. The relationship between the higher level entity and its subtypes is expressed in terms of the attribute which makes each occurrence a member of a specific subtype, rather than in terms of some verb. In our animal example, we might use attributes such as "method of body temperature regulation", "diet", or "number of feet" to categorize the animal entity into subtypes. Doctor, Nurse, and Technician are related to the supertype Employee by the attribute "occupation" and this relationship is expressed by the attribute "occupation" rather than as a verb. (Note: In a payroll function this distinction might not be sufficient to justify partitioning into subtypes. In a hospital function such partitioning would probably be appropriate, as Doctors, Nurses, and Technicians all behave differently in the hospital business. The decision whether to categorize into subtypes will be based on the groups understanding of the function.)

For purposes of definition, we will refer to this second type of relationship as a "subtype relationship." All other relationships will simply be referred to as "relationship". This document will make few references to subtype relationships.

Where difficulty is encountered in expressing a relationship, the situation should be examined closely. It may be that multiple relationships exist between two entities and should be expressed as such. It may be that the true relationship is with an as yet unidentified entity subtype, in which case that entity subtype should be identified. It might also be that the relationship is a complex relationship.

Thus far we have talked exclusively about binary relationships, or simple relationships between two entities. There are also complex relationships. These are relationship that exist among three or more entities. They are sometimes referred to as "n-ary" relationships. The relationship among a doctor, a patient and a surgical procedure can be expressed as three binary relationships (doctor to patient, patient to surgical procedure, and doctor to surgical procedure). This same relationship can be more meaningfully expressed as a complex (ternary, or n-ary) relationship. Specific techniques for addressing complex relationships will be addressed in formal training.

Relationships between entities should be established at the highest appropriate level. The "is employed by" relationship from Organization should be made to the supertype Employee rather than to each of the subtypes Doctor, Nurse, and Technician. The "provides primary care" relationship from Patient, however, should be established directly to the subtype Doctor.

As relationships are identified they will generally be between entities that clearly belong within the scope of the function being documented. Some relationships may involve an entity which just as clearly belongs in another business function. Such relationships should be documented and the entities in question represented on the entity relationship diagram developed in Task 3.1.2.6. However, rather than create descriptions of those entities which belong within the scope of other functions, the descriptions developed by those functions can be used. Resolution of potential inconsistencies in this area will be accomplished as a natural part of the ongoing integration process.

All relationships should be documented with a narrative that meaningfully describes the relationship.

TASK 3.1.2.5 Document Business Rules

Purpose: To document business rules identified in the data model. Business rules may result from the constraints placed on relationships by business practices.

Outputs:

- o Business rules

Relationships: Reference will be made to the business practices identified in Task 2.1.6.2. Business rules will become part of the future information requirements for the function.

Approach: Business rules describe characteristics of the entities and their relationships that reflect the business practices of the function. Business rules may apply either to the entities themselves or to the relationships among entities. Application of business rules preserves the integrity of the data.

The constraints placed on relationships due to business practices identify one category of business rule. For example, it may be that a class <u>must</u> be attended by students (that is, a given instance of class must be related to at least one instance of student), but that a student might not have attended any classes (an instance of student might not be related to any specific instances of class). Such constraints could be more explicitly expressed as "a student may attend from zero to any number of classes, and a class must be attended by at least 6, but not more than 30 students." Such a statement of the relationship with the associated entities and constraints constitutes one category of business rule which is identified during the data modeling effort. This rule reflects the business practice of canceling classes for fewer than 6 students and of not registering more than 30 students to keep classes at a manageable size.

Some sample business rules are:

"The Customer Account Number on the Customer Order must belong to a Customer in the Customer file."

"A Customer Record cannot be deleted until all outstanding Invoices for that Customer are paid."

"The effective date of a promotion action cannot be earlier than the current processing date."

Note in particular that the last rule would prohibit processing retroactive promotions. This reflects one choice of business practice from among many alternatives. The key to writing effective business rules is to accurately capture the practices that are most appropriate for the function being studied.

The establishment of candidate identifiers also represents a type of business rule. Where Invoice Number is established as an identifier of the Invoice entity, it represents the business rule that no two Invoices may have the same Invoice Number.

Business rules will be captured and described as part of the data model. They may be expressed in the form of relationships, candidate identifiers, or simply as narrative statements.

Task 3.1.2.6 Develop Entity Relationship Diagram (ERD)

Purpose: To develop the entity relationship diagram.

Outputs:

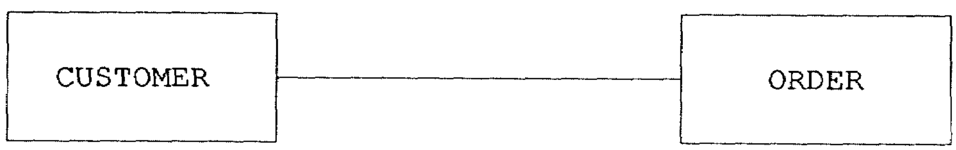
- o Entity relationship diagram

Relationships: The entity relationship diagram will become part of the future information requirements of the function.

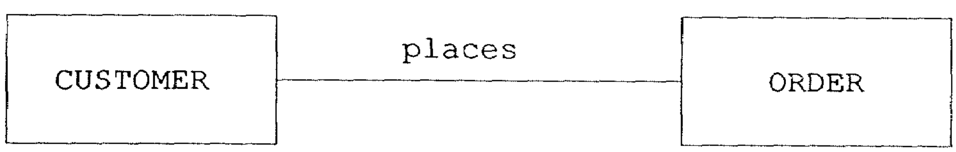
Approach: The entity relationship diagram is created by drawing a graphic representation of entities and entity subtypes (shown as boxes) connected by relationships (represented as interconnecting lines). Detailed conventions for creating entity relationship diagrams will be covered in formal training. Brief descriptions of the symbology and sample documents are shown on the following pages.

Entity Symbol - A rectangular box used to represent an entity on the diagram and containing the name of that entity.

Relationship Symbol - A straight line between two entities representing a relationship between those entities.

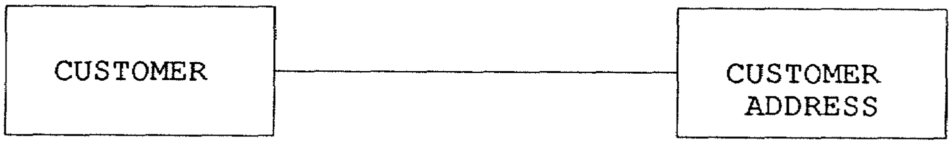


Relationship Name - A name indicating the nature of the relationship. The name will normally consist of a verb or verb phrase (in the case of a relationship between an entity and its subtypes the relationship will be expressed in the form of the criteria which is used to categorize the subtypes). The relationship between Customer and Order might be called "places" to indicate that the Customer places Orders.



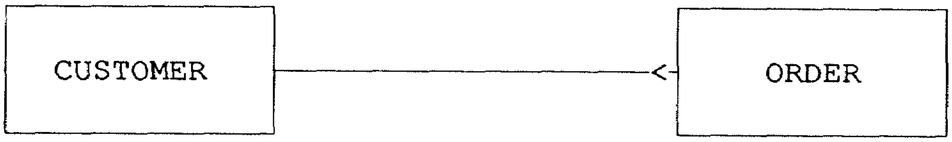
Cardinality Indicator - A measure of the number of instances of one entity that are related to a second entity. For example, not only are Customers related to orders, but a single customer may be related to multiple orders. Cardinality is shown on an ERD by the use of the "crowfoot" on either end of the relationship symbol.

Cardinality relationships can be "one to one" (Customer to Customer Address), "one to many" (Customer to Order), or "many to many" (Order to Product). "Many to many" relationships are generally resolved to two separate "one to many" relationships by the creation of a new intermediate entity.



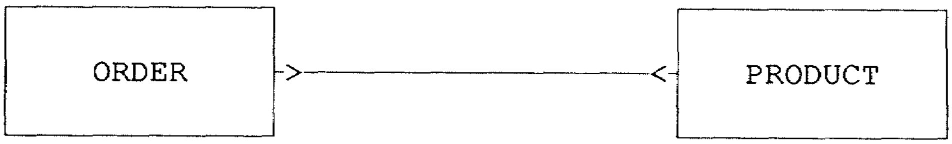
One to One

A given Customer is related to one Customer Address, and each Customer Address is related to one Customer.



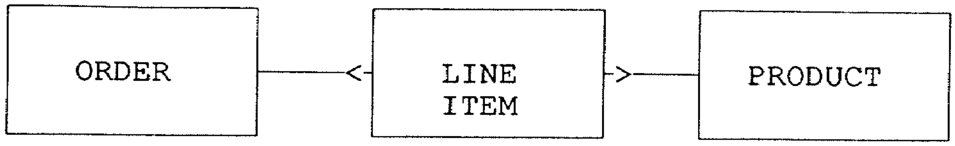
One to Many

A given Customer may be related to many Orders, but each Order is related to only one Customer.



Many to Many
(unresolved)

A given Order may be related to many Products, and a given Product may be related to many Orders.

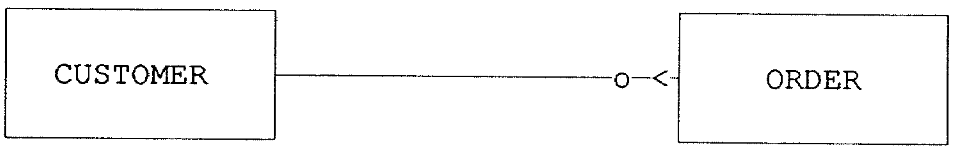


Many to Many
(resolved)

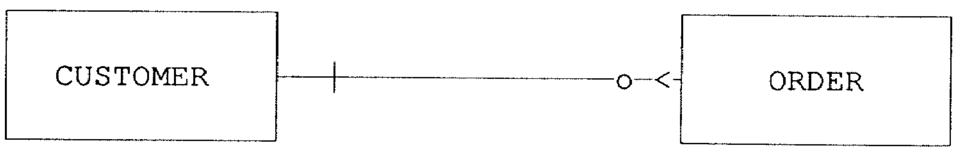
A given Order may be related to many Line Items, and a given Line Item may be related to only one Order. At the same time, a given Product may be related to many Line Items, and a given Line Item may be related to only one Product.

Cardinality can also be used to express business rules. The example above suggested that an Order could be related to many Line Items. An alternative business rule would be that an Order can be for only one Product (the business deals only in big ticket items and chooses to do business this way). In this case the cardinality should be expressed accordingly.

Optional Relationship Indicator - Indicates that a single instance of one entity may or may not be related to any instances of a second entity. An optional relationship is indicated by a small circle on the end of the relationship line closest to the entity that is optional. The following example illustrates a situation which allows for new Customers who might not be related to any Orders.

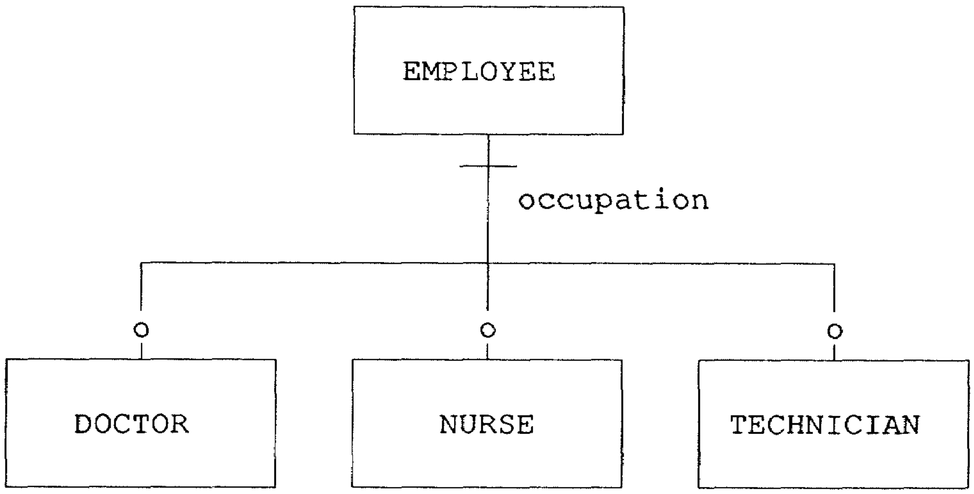


Mandatory Relationship Indicator - Indicates that an instance of one entity cannot exist without a relationship to at least one instance of a second entity. A mandatory relationship is indicated by a bar which intersects the relationship line and is perpendicular to it. The example below shows that while a Customer may not be related to any Orders, it makes no sense to have an Order that is not related to a Customer.

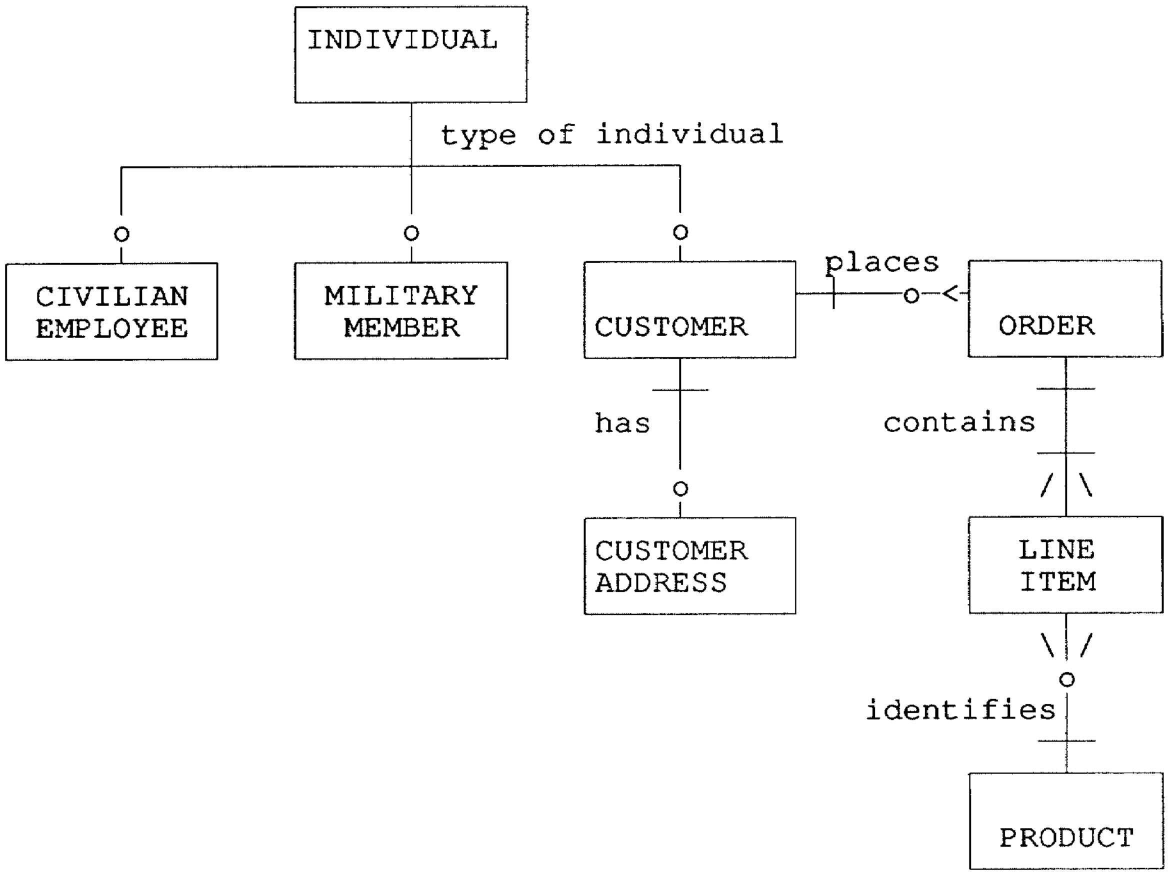


Subtype Relationship - A relationship between levels in an entity/entity subtype hierarchy is expressed as a branching line which extends from the higher level of the hierarchy and branches out to each of the lower level subtypes.

When examining subtype relationships each instance at the higher level can belong to only one of the lower level categories. It would be normal to see the mandatory indicator next to the higher level, indicating that each entity subtype must be a subtype of the higher level entity. One would also expect to see the optional indicator next to each of the lower level subtypes, indicating that the higher level may belong to any one of the subtypes.



Entity Relationship Diagram - The following ERD shows all of the symbology we have described on a single sample document.



The ERD shows an entity/entity subtype hierarchy of Individual categorized as Civilian Employee, Military Member, or Customer based on the attribute "type of individual."

The subtype Customer is, in turn, related to the entity Customer Address. A Customer may or may not "have" a Customer Address, but a Customer Address must be related to a Customer.

Customers may "place" any number of Orders (or none at all) and Orders must be placed by a Customer. Orders must "contain" some number of Line Items. Each Line Item "identifies" exactly one Product, but a Product may be identified by none or any number of Line Items.

Task 3.1.2.7 Integrate Data Across Functions and Paths

Purpose: To check cross-functional data definitions and requirements.

Outputs:

- o Revised entity/entity subtype structures
- o Revised attribute names and definitions
- o Updates to the Corporate Data Dictionary

Approach: The approach to defining DoD's corporate information systems requirements includes subdivision by function at the highest level. A second subdivision was made by approaching definition of these functions across three parallel paths: the Future Functional Path, the Current (or Composite) Functional Path, and the Existing Information System Path.

While the processes for each function are generally unique to that function, and while they may further vary across paths, the data which these processes manage is shared across the corporation. The Individual Social Security Account Number, for example, is the same data element regardless of whether the function using it is Payroll or Contract Management. It is also the same whether we are talking about the future function, the current composite function, or an existing information system. As we progress through the data modeling effort we must ensure that shared data is described consistently throughout DoD. In this task, the Functional Groups data administrator reviews the identified data requirements with the information architecture staff to ensure cross-functional data integrity.

To achieve effective data management across the organization, we must first achieve effective management of metadata to avoid redundant metadata descriptions and the need to retrofit metadata.

The definitions and descriptions of data are commonly referred to as "metadata." While the data is not owned, metadata is. Metadata is owned by an authoritative data steward. The data steward is the individual in an organization who is considered expert in the data and is capable of authoritatively defining the metadata for the particular data in question. The same data steward has responsibility for a given data attribute both across functions and across paths. Generally the data steward is involved with the business function which is responsible for the existence of the data.

It is only in this context that the steward owns the data. While the entire organization can share the data itself, everyone must use the metadata defined by the owner, or data steward.

Included in the category of metadata are data attribute definitions and entity subtype structures. The data steward for each entity should be identified as early as possible in the data modeling process.

A critical step in effectively managing metadata has already been described by the use of a standard Corporate Data Dictionary. By defining data attributes in a common dictionary, using common naming standards, and placing the data dictionary under the control of a corporate data administrator, we have assured that each function views the data consistently.

Data attribute names and definitions must be consistent both across functions and across paths. This cannot be accomplished without the use of an automated and centralized Corporate Data Dictionary.

It will not be unusual to discover that some data attributes have names which preexist the data modeling effort. Individual Social Security Account Number has been commonly referred to as SSN since long before this effort began. Such preexisting names will be particularly common in the composite function path and the existing information systems path. Where such attribute names do not meet current naming standards, a new name must be created. The old name can then be established in the dictionary as an alias for the standard name. Aliases do not have their own definitions, but can be used to reference the standard definition.

A second area where consistency must be maintained is in the creation of entity/entity subtype structures. In our earlier example we categorized Individuals as Civilian Employees, Military Members, and Customers. Individual could just as easily be categorized as "living or dead", "male or female", or by state of birth. Which method of categorization is most appropriate depends upon the perspective of the function being described. Different functions may have different perspectives, and therefore, would tend to create different subtype structures. However, it is the "owner" of the data who must make the determination as to which structure is most appropriate for corporate-wide use.

For example, if we determine that the personnel function owns Individual, it is personnel which must define the entity subtype structure. If the financial business function needs to use the data, that function must use the structure defined by personnel and cannot create its own. If the financial group feels that their information requirements cannot be satisfied by the structure provided, a solution must be negotiated with personnel. Financial will then use the subtype structure defined by Personnel in their Financial data model.

Identification of shared entities is accomplished during integration activities. Entities are generally shared in the sense that an entity is managed (created) by one function and referenced by another. Shared entities will be part of the external interface of the referencing function. Where shared entities are discovered, the group must investigate the function which owns the entity and adopt subtype structures as appropriate. Owning functions may be identified by tracing entities via the external interfaces through which the data is acquired by the function.

Entity/entity subtype structures must be consistent among functions within a path. However, unlike data elements, these structures do not have to be consistent across paths (that is, between the future model and the composite models for the same function). Since these structures represent a way of looking at the organization of data from a functional perspective, and since each path may describe a different functionality (the future function may not be the same as the current function), it is acceptable that these structures be different across paths.

Most overlaps among functional groups will be identified during integration activities and will be resolved through negotiation.

Duplicating entities and entity subtype structures and their descriptions in multiple data models as we have described is recognized as being redundant. However, to fully represent the information requirements of individual functions, such redundancy is necessary. Redundancy of the effort involved in creating these definitions is <u>not</u> necessary. Where one function uses data owned by another, that function should reference the metadata documented by the owning function rather than expend resources in creating its own.

The Functional Groups data administrator will help the group identifyy any cross-functional inconsistencies in data attribute names, definitions, or any other elements of the data model. If inconsistencies exist, the group will revise the entity/entity subtype structure and attribute names and definitions. Any updates to the Corporate Data Dictionary will be made at this time.

STEP 3.1.3 FUTURE HIGH-LEVEL MODEL RECONCILIATION

Purpose: To reconcile the action diagrams and supporting information views with the high-level data model for the purpose of ensuring the completeness of the two models and of maintaining consistency between them.

Description: In this step the entities and entity subtypes identified in the data model from Step 3.1.2 will be reconciled with the action diagrams from Step 3.1.1.

A subprocess to entity CRUD matrix will be developed. Aggregated action diagram information views will be compared to the matrix to ensure that the data model satisfies each information processing requirement at the subprocess level.

Outputs:

- o Subprocess level information views
- o Subprocess to entity (CRUD) matrix
- o Revised data model including
  - o New entities and entity subtypes
  - o New entity and entity subtype descriptions
  - o New entity and entity subtype relationships
  - o Revised entity relationship diagram
  - o Newly identified attributes with descriptions
- o Revised process model
- o Updates to the Corporate Data Dictionary

Relationships: The data model that is output from this step will facilitate an evaluation of the ability of existing systems to meet future and composite information needs and can be used as a basis for strategy decisions regarding transition from composite to future functions.

The revised data model will be fully attributed in Step 3.1.5. The revised process module will be described to the module level in Step 3.1.4.

Task 3.1.3.1 Aggregate Action Diagrams to Subprocess Level Information Views

Purpose: To aggregate the action diagram level information views into subprocess level information views.

In Step 3.1.1 subprocesses from Phase II were partitioned into macro-level action diagrams and an information view was developed for each action diagram. The overall purpose of Step 3.1.3 is to ensure that the information requirements of the action diagrams can be satisfied by the data model. The specific purpose of this first task in that step is to consolidate the number of information requirements into a manageable number.

Outputs:

- Subprocess level information views

Relationships: Aggregated subprocess level information views will be used to build the CRUD matrix in Task 3.1.3.2.

Approach: Given the significant number of action diagrams which may be needed to represent a large system, creating an entry on a CRUD matrix at the level of action diagram would result in a document of unmanageable size. For this reason the information views will be aggregated back to the level of subprocess (as identified in Phase II). The process model entries for Task 3.1.3.2 will then be at the level of subprocess. The corresponding view will represent the sum of the information views for all of the action diagrams within a given subprocess.

For example, assume that we identify three action diagrams for a subprocess which we will call Subprocess One. In turn we would identify an information view for each action diagram. The view for Action Diagram 1 might be a reference to Entity 1. The view for Action Diagram 2 could consist of a reference to Entity 2 and an update of Entity 3. Finally, the view for Action Diagram 3 might be to create, reference and delete Entity 3. These three views could be aggregated to form a single view consisting of references to Entities 1 and 2 and a create, reference, update, and delete of Entity 3. A data model which satisfies the aggregate view would, by definition, satisfy the individual Action Diagram views.

It may appear that the effort to identify action diagram level views, only to aggregate them later, was redundant. In fact this detailed understanding of information requirements is necessary to fully understand the data requirements of the subprocesses. Further, the detailed action diagram level views will be used later in Step 3.1.4.

Task 3.1.3.2 Develop Subprocess to Entity (CRUD) Matrix

Purpose: To facilitate the validation of the data model against the process model by showing on a single document all entities and entity subtypes arrayed with all subprocesses.

Outputs:

- o Subprocess to entity (CRUD) matrix

Relationships: The CRUD matrix will be used to validate information views in Task 3.1.3.3 and to validate the management of entities in Task 3.1.3.4.

Approach: A matrix is created showing subprocesses from Task 3.1.3.1 (aggregated action diagrams from Step 3.1.1) on the vertical axis and data entities from Step 3.1.2 on the horizontal axis.

The nature of the relationships between subprocesses and entities are then expressed in the context of the action(s) that the subprocess may take on each entity. Potential actions are:

C Create: subprocess creates instances of the entity

R Read or Reference: subprocess references instances of the entity

U Update: subprocess modifies attributes of an instance of the entity

D Delete: subprocess may remove instances of the entity

For each intersection of a subprocess and an entity on the matrix the group must consider potential action(s). If there is no action the intersection remains blank. Otherwise one or more characters are entered at the intersection as appropriate.

Since this Matrix is critical to the completion of subsequent tasks in this step, we will briefly discuss interpretation of the following sample matrix.

Sample Subprocess to Entity (CRUD) Matrix

(Ent = Entity, C = Create, R = Reference, U = Update, D = Delete)

	Ent 1	Ent 2	Ent 3	Ent 4	Ent 5	Ent 6	Ent 7	Ent 8
Subprocess 1	C							
Subprocess 2		R		R		R	R	
Subprocess 3			CRUD					
Subprocess 4							CUD	
Subprocess 5		R		CRUD				
Subprocess 6								
Subprocess 7		R				U	R	
Subprocess 8	R	CRUD						R
Subprocess 9							R	

The horizontal line extending to the right from each subprocess is one representation of the information view of that subprocess. In the sample CRUD matrix above the view of Subprocess 2 consists of references to Entities 2, 4, 6, and 7. The information view of Subprocess 7 includes reference to Entities 2 and 7 and the potential updating of Entity 6.

The vertical line extending beneath each entity represents management of that entity within the function. Entity 2 is created, read, updated and deleted by Subprocess 8 and merely referenced by Subprocesses 2, 5, and 7.

Columns that contain only "R"s indicate that the function being documented only references the data and does not manage it beyond that. In such cases the entity is being managed by some other function, i.e., the data is part of the external interface of the function being modeled.

The information view for Subprocess 6, which does not act on any entities, is an obvious problem. Virtually every subprocess will have some type of information view. Either the data requirements of the subprocess are not properly understood, the information view is not needed, or another entity needs to be added to the matrix. Entities will be added only after careful verification that the data model is incomplete. First consideration will be given to potential scoping problems. It may be that the process in question exists in the current function, but may belong in another function in the future. In this case the process itself will be removed and transferred to the appropriate business function.

Similarly, the column under Entity 5, which shows that no subprocesses act on Entity 5, needs to be investigated. If the data entity is not referenced by any subprocesses within the function, then either there is no need for that entity to be included in the data model for this function, or a subprocess which does manage the entity has been overlooked and must be added to the process model. Again, careful consideration must be given before removing entities from the data model. The group must ensure that the entity is not needed now or will not be needed in the future by some as yet undocumented process. Perhaps the expanded scope of the future function was not properly considered in the process model.

Removal of an entity from the data model of one particular function does not imply that it does not continue to exist on data models of other functions.

Discrepancies such as those described above are a natural part of the reconciliation process. They are simply one step in the reconciliation of the process and data modeling efforts which have proceeded along parallel, but independent, paths.

Since this subprocess to entity matrix will be of considerable size and may require considerable manual effort, automated support is considered essential.

While our matrix tells us which entities are needed to satisfy the information requirements of each subprocess, it does not resolve any questions we may have at the level of data attributes. To be fully understood, information views must be expressed at the attribute, or data element, level. However, we have not yet fully defined our data to the attribute level nor have the information needs of our subprocesses been consistently identified in such detail. This level of validation must wait until Step 3.1.6.

Task 3.1.3.3 Validate Information Views Against the Data Model

Purpose: To ensure that each of the information views identified in process model action diagrams can be satisfied by the data model.

Outputs:

- o Revised data model including
  - o New entities and entity subtypes
  - o New entity and entity subtype descriptions
  - o New entity and entity subtype relationships
  - o Revised entity relationship diagram
  - o Newly identified attributes and descriptions
- o Revised process model
- o Updates to the Corporate Data Dictionary

Relationships: Validated information views will become part of the future process model and will be used in Step 3.1.5 to develop detailed information views (entity attributes). The validated (normalized) data model will be fully described in Step 3.1.5.

Approach: In Step 3.1.1 the group developed an information view for each action diagram. These information views were defined from a process perspective and reflect the data needed by the associated action diagram to execute its task. While in some cases individual data elements may be included in the information views, the data model itself is not yet fully attributed and views may be reliably validated only to the entity level.

Validation of views may be accomplished by examining the horizontal line extending from each subprocess. If the CRUD indicators match those defined in the information view, no action is necessary. Where they do not match, it is an indication that corrective action may be necessary.

The group will first verify that the information view and the data model terminology is expressed consistently and that the problem is not simply one of semantics. Strict application of the corporate naming standards should eliminate the potential for this problem.

The group will then reexamine the subprocess requiring the information view and ensure that it is within the scope of the future business function as defined in Step 1.1.1. If it is not, the subprocess will be added to the process model for the appropriate business function and removed from the business function in question.

Where the information view is appropriate and correct, and is not satisfied by the data model, it is an indication that the data model itself is incomplete. In this case the data modelers will take appropriate action to add any needed entities and entity subtypes, complete with relationships and descriptions, to the data model.

Task 3.1.3.4 Verify Entity Management

Purpose: To ensure that all entities are created, referenced, updated and deleted as appropriate within the function.

Outputs:

- o Revised data model
- o Revised process model
- o Revised subprocess to entity (CRUD) matrix

Relationships: The data model from Step 3.1.2 and the process model from Step 3.1.1 will be validated in this step. These models will be fully described in Steps 3.1.4 and 3.1.5.

Approach: The CRUD matrix will be examined to ensure that all entities are "managed" appropriately. That is, the group will verify that processes exist which appropriately create, reference, update and delete each of the entities in the data model. Entity management problems must be resolved. If the Materiel Management function needs to reference the Item entity, and no process can be identified which creates Items, then there will be no Items to reference. Entity management may occur either in the function being studied, or in some other function. Verification is accomplished by examining the vertical columns extending beneath each entity.

If there are no indicators in a column it is evidence that the entity may not be needed by the function or that a process within the function has been overlooked. The group must carefully consider the future scope before removing any entities from the data model. Care must be taken to ensure that no processes have been overlooked. If no process can be discovered that at least references the entity, the entity should be removed from this data model and perhaps added to the data model for another function.

If there are no create indicators in a column it indicates that the entity is imported from an external function. This will be verified by coordination with the external source of the data to ensure that the required data will be available to support the function when needed. The group must ensure that a process to create the entity already exists or will be established either in this or some other business function.

STEP 3.1.4 FUTURE DETAILED PROCESS MODEL

Purpose: To refine the macro-level action diagram into a detailed action diagram and clearly represent processing control structures, data actions, and data elements. The output will provide the system designers with a detailed view of the processing requirements of the business function.

Description: This step develops a detailed action diagram by reviewing the macro-level action diagram and adding process control symbols from the expanded action diagramming repertoire. Through the inclusion of these additional action diagramming symbols, emphasis is placed upon identifying major processing modules, opportunities for concurrence, data actions, and basic data elements. As these processing control structures are introduced, a series of business practices will be documented which detail "how" to perform processing.

- o A business practice is an expression of "how" to manage and execute an aspect of a functional activity.

Outputs:

- o Detailed action diagram showing data actions
- o List of data elements
- o List of business practices

Relationships: The basis for this step is the macro-level action diagram produced by Task 3.1.1.1. The action diagram produced in this step will be further refined in Step 3.1.6. The list of data elements will be integrated with the list of data attributes generated by the data modeling group. This process to attribute matrix will assure data integration across the process and data modeling analyses. After the data integration subtask a consolidation review with the other functional groups is recommended.

Task 3.1.4.1 Future Process Flow Analysis

Purpose: To further refine the macro-level action diagram to reflect sequence control and procedures.

Outputs:

- o Action diagram showing detailed process logic

Relationships: The basis for this step is the macro-level action diagram produced by Task 3.1.1.1. The action diagram produced in this step will be further refined in Step 3.1.6.

Approach: This task further refines the macro-level action diagram into a detailed action diagram. This is done by reviewing the macro-level action diagram and adding process control symbols from the expanded action diagramming repertoire. Emphasis is placed upon identifying major processing procedures (called modules), opportunities for concurrence, and shared processing modules.

The action diagram will be enhanced by adding the concept of modules and concurrence flow control. Figure 1, below, graphically illustrates the representation of each of these diagramming constructs. The macro-level action diagram produced by Task 3.1.1.1 is reviewed in two passes to identify processing logic details. The first pass emphasizes the identification and categorization of the action modules. The second pass looks for opportunities for concurrent processing and direct changes in processing flow.

Three different types of modules should be identified in this step.

- o A unique processing module is an activity component of an action which does not occur elsewhere in the function or only occurs elsewhere with significant processing changes (as in differing service business practices). For example, a module to "produce pay check."

- o An undefined processing module is a collection of unknown (by the work group) processing activities which will be defined by the design group at a later date. For example, a module to "electronically send contract data."

- o A common processing module is a recurring set of processing activities which are used in several locations in performing the processing function. For example, a module to "perform error handling."

The concurrence tool affects the processing flow. It can be used to simplify the logic and may allow the designer some additional flexibility when trying to streamline the processing.

- Actions and modules which should be considered for concurrent processing are those which do not have any inherent sequential order of precedence relative to the surrounding processing actions, i.e., actions or blocks of actions which can be accomplished at the same time as some other action or block of actions.

Subtask: Identify Processing Modules

The macro-level action diagram must be reviewed to identify unique, undefined, and common processing modules. Previously the actions comprising the process or subprocess were identified. Often these actions either represent a set of processing activities and are already modules or can be logically grouped and replaced with a module. If grouping common sets of actions simplifies the logic, use of modules (which should be defined as specifically as possible in the documentation) is justified.

As modules are identified or created, a description containing a mini-action diagram and narrative explanation should be prepared. Once the macro-level action diagram has been defined at the module level it is referred to simply as an action diagram. This terminology will be used during the next final stages of process decomposition.

Subtask: Identify the Sequence Control Structure

Having identified the processing steps to the module level in the action diagram, it is now possible to analyze opportunities for concurrent processing and restructuring of the process flow.

Not all opportunities for concurrence need to be taken or identified. Attempts to increase concurrence may disrupt the underlying logical flow of the process, causing needless complexity. An opportunity for concurrence is a concurrent process which does not disrupt the logical flow of the function process. For example, in the Civilian Pay function, the process "calculate deductions" may require sub-processes of (1) identify health benefits deductions, (2) identify Combined Federal Campaign donations, and (3) identify life insurance allotments. If these can be accomplished in parallel, they could be identified as concurrence opportunities. Concurrence should be viewed as a way of streamlining the overall function processing and thereby simplifying its structure.

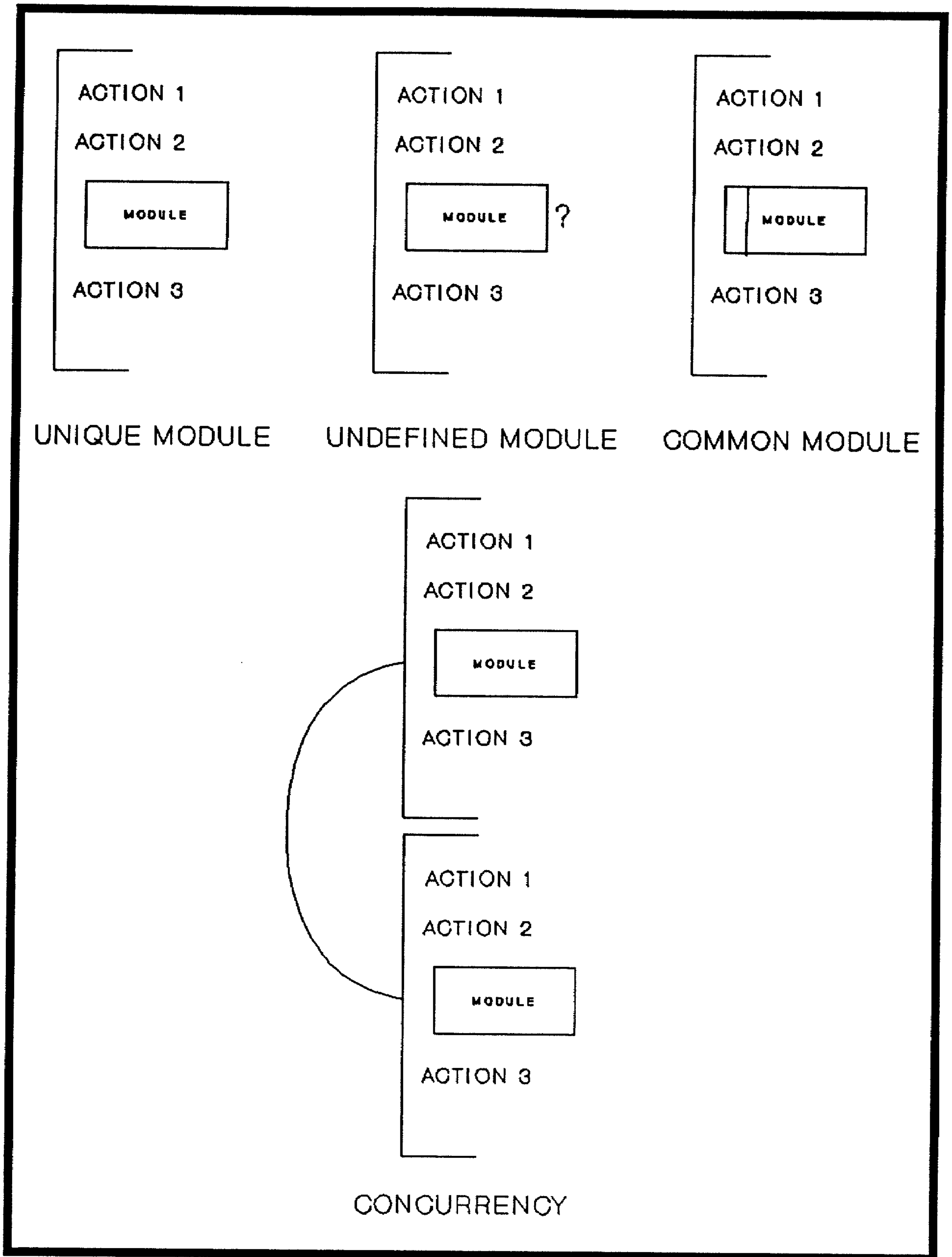


Figure 1 Process Control

Task 3.1.4.2 Detailed Future Information Processing Analysis

Purpose: To identify the detailed data actions performed at the module level. It will identify processes which use data, and where appropriate the method of use.

Outputs:

- o Detailed action diagram showing data actions
- o List of data elements
- o List of business practices

Relationships: The action diagram with all modules identified was produced by Task 3.1.4.1 and serves as the primary input to this task. The list of data elements will be used in Step 3.1.6.

Approach: This task adds the simple and compound data actions to the action diagram to further depict the input and output activities within the function. This is the most detailed level of action diagram produced in the analysis of the function.

The action diagram produced by Task 3.1.4.1 is reviewed in two subtasks to identify input and output processing details. The first pass emphasizes the identification of simple data actions and the second subtask emphasizes compound data actions.

- o Simple data actions are elementary data activities which include Create, Read, Update, and Delete (CRUD) processing. In the action diagram they are represented by a line containing one of the CRUD key words followed by a single line box identifying the information view being acted upon.

- o Compound data actions are complex data activities which for this analysis will only include SORT and JOIN. In the action diagram they are represented by a compound data action key word, followed by a double line box containing the information view being acted upon, followed by a condition phrase such as BY, IF, or WHEN.

The data actions are extensions to the action diagram symbols repertoire and completes the set of diagramming symbols. Figure 2, below, graphically depicts these symbols. Completion of this level of action diagramming represents the final step of the process decomposition methodology.

After bringing the action diagram to the final level of detail, the data element list produced by Task 3.1.2.3 will be reviewed and updated. This updated list represents the maximum level of data element detail attainable by the process work group and will be integrated with the analysis performed by the data modeling group in the next step.

Subtask: Identify Data Actions

The data diagram should be reviewed for actions which are or require data activities. Typically, the simple CRUD data actions will suffice. However, some processes will require compound data actions. These occurrences should be carefully considered to eliminate design-like influences from the inherent, processing required instances. It is critical to only identify the "what" aspects of the data activities and to resist specifying any "how" design approaches. Most often this analysis will involve the conversion of already identified actions and modules to a more specific data activity structure, such as changing READ CONTRACT-STUFF to SORT CONTRACT-STUFF BY DATE-RECEIVED.

It is not unusual for some confusion to arise over the SORT and JOIN verbs as used in this type of analysis, when compared with their use in design. For this analysis effort <u>the SORT action identifies occasions when the data inherently must be in a specific order.</u> It is not an attempt to influence the flow of processing, but rather a statement of a requirement. In the contract example above the contracts might require processing in date received order because of a business practice - <u>not</u> because it is more efficient or easier.

<u>The JOIN verb identifies processing actions which require data external to the function being analyzed.</u> For example, if the Contracts Payment analysis has a process requiring data maintained by the Contracts Management area, a JOIN could be used to signify this requirement. It could look like JOIN AUTHORIZED-PAYMENT WITH VENDOR-ADDRESSES BY VENDOR-IDENTIFIER. In this example, AUTHORIZED-PAYMENT is known to exist within the Contracts Payment area; however, the VENDOR-ADDRESSES are part of the Contract Management area. The JOIN signifies that this process requires data controlled by another function.

The use of SORT and JOIN symbols is intended to add to the information being collected and depicted in the analysis. As their application is not always required or not always obvious, care must be taken to assure that they are not being used as design symbols. SORT and JOIN can provide details on business practice requirements and integration requirements when used in the analysis. They typically will act upon information views but also can identify data elements performing special roles like keys (data controlling record order) or indices (data for linking or looking up related data). This compound data action will be useful to the designers responsible for creating the final system.

Subtask: Generate a List of Process Required Data Elements

After completing the data action decomposition, the action diagram should be reviewed for modifications and additions to the data element list created by Task 3.1.1.2. The addition of the data action structures to the diagram should serve to clarify some of the information requirements and data dependencies in the process. These insights may cause new or modified information views to be created and new data elements to be identified. These elements should be added to the Task 3.1.1.2 list for use in the Step 3.1.6 reconciliation analysis.

Subtask: Generate a List of Business Practices

As the action diagram gains more and more detail, the methods and techniques for accomplishing processing requirements will become discussion issues and will require documentation. These are the business practices of the function. They influence the sequence and control of the processing and define the normal and exception logic control conditions. As each business practice is identified, it should be documented through a brief descriptive narrative. A name or short phrase identifying the business practice should be created for ease of reference, and a comprehensive list generated.

Business practices were introduced in Task 2.1.6.2 and preliminary definitions provided. This documentation should be reviewed in light of the additional processing details now available to the work group. Business practices from Task 2.1.6.2 should be further clarified, and any additional business practices should be added to the list. For consistency the format identified in Task 2.1.6.2 should be continued.

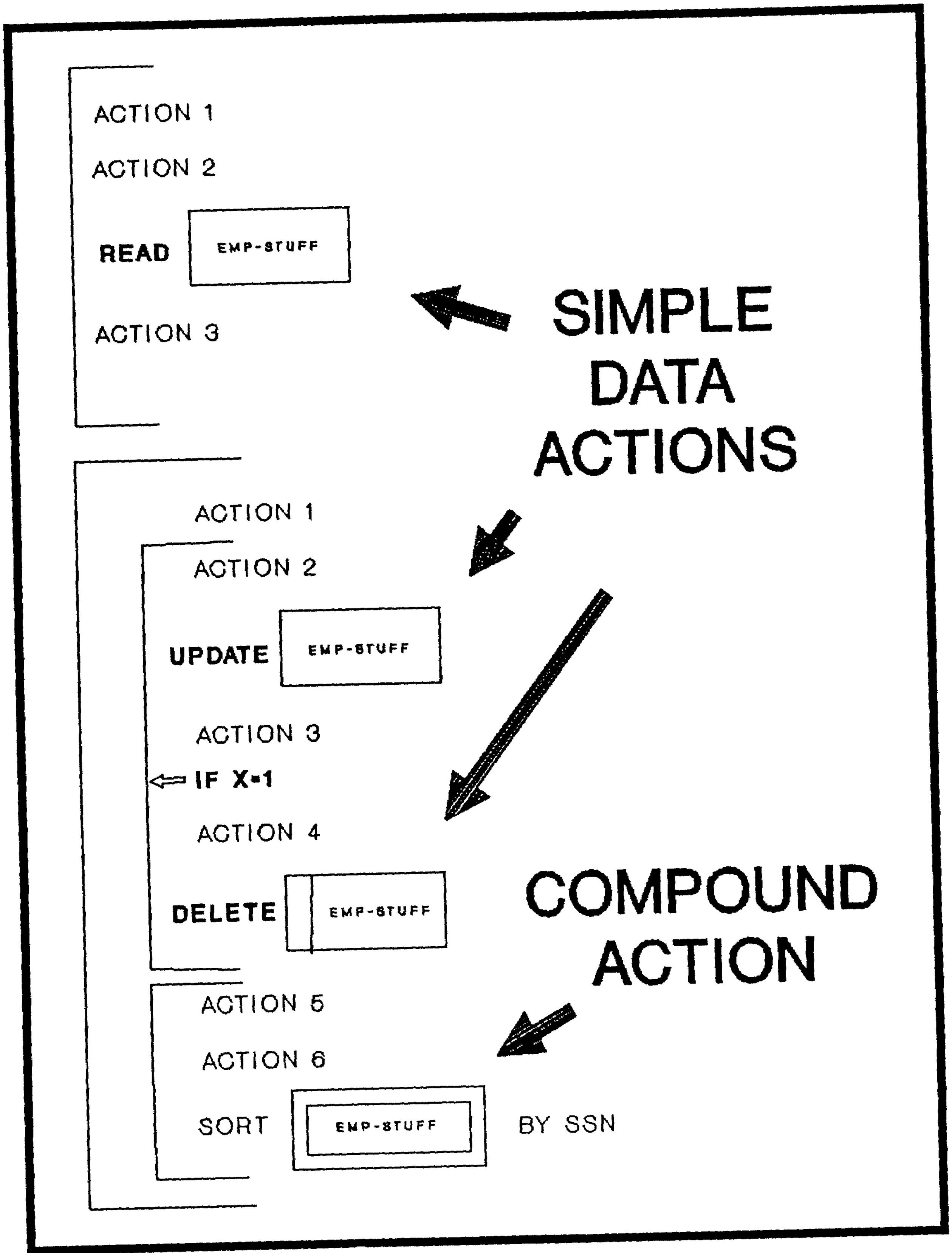


Figure 2 Data Action

STEP 3.1.5 FUTURE DETAILED DATA MODEL

Purpose: To describe at a detailed level the data portion of the future information requirements for the function.

Description: In Step 3.1.2 a high-level future data model was created. In 3.1.3, this model was reconciled with the logical process model (entity diagram) from Step 3.1.1. In this step the data model will be fully attributed and carried to its final level of detail. The data model will then be normalized.

In the Phase III data modeling process the data will not be organized into independent application oriented subsets. However, the data will be organized such that it lends itself to partitioning into such subsets for the purpose of satisfying the information requirements of a variety of implementation strategies. This partitioning is not a functional group activity. It will be performed during transition to future standard information systems.

Outputs:

- Attribute names and definitions
- Updates to the Corporate Data Dictionary
- Normalized data model
- New entities and entity subtypes, with descriptions

Relationships: The detailed data model will be used by central design activities to design physical data structures in support of applications.

Task 3.1.5.1 Identify Attributes

Purpose: To identify all additional attributes needed to fully describe the entities represented on the data model.

Outputs:

- o Attribute names and definitions
- o Updates to the Corporate Data Dictionary

Relationships: The attribute definitions will be entered in the data dictionary for reference by functional groups, central design activities, and other interested parties throughout the life cycle of the business function.

Approach: In Step 3.1.2 the group identified a limited number of attributes, including candidate identifiers. Identification of these attributes was as a fall out of the modeling process, not as part of a concerted effort to fully attribute the date model. In this step the group will make such an effort and will identify all additional attributes needed to fully describe the entities represented on the data model. In Step 3.1.2 the group was also encouraged to minimize the amount of information captured about each attribute. In this step the group will collect all relevant information. The end product of this subtask will be a set of fully attributed entities.

As in the previous step we will follow the principle of inheritance which allows us to establish attributes at the highest level of a hierarchy of entities and entity subtypes. These attributes are then inherited by each of the subtypes.

All attributes will be named in accordance with corporate naming standards and entered into the Corporate Data Dictionary.

As development proceeds, it will be common to find that metadata (names and definitions) for attributes identified in the function being studied has already been created by other functional groups. In many cases this metadata may not agree with what the group would like to use. However, it is critical to the integrity of the corporate data of DoD that the group not create its own names and definitions. Sound data management principles call for the authoritative data steward to provide metadata for each attribute. Where metadata is in dispute, the appropriate data steward will be identified and consulted.

While the data steward "owns" the metadata, the data administrator is the final authority in controlling entries into the Corporate Data Dictionary. The data administrator ensures that appropriate naming conventions have been followed and that the integrity of the data in the dictionary has not been compromised. In cases where the group may function as the data steward for selected pieces of data, they must still bow to the authority of the data administrator on data standardization and integrity issues.

The group must not create local names and definitions for specific use within a function for data which is shared across functions. Rather, the group must use legitimate corporate metadata created by others and will create metadata only for those attributes for which they are the data steward.

Attribute information will include such things as the attribute name, its description, purpose, domain values, default values, optionality, security, length, type, and ownership.

Task 3.1.5.2 Normalize the Data Model

Purpose: To further standardize the structure of the data by applying the formal rules of normalization. The process of normalization will ensure that the data model is consistent, non-redundant, stable, and free of process bias.

Outputs:

- o Normalized data model
- o New entities and entity subtypes, with descriptions
- o New attributes and definitions
- o Updates to the Corporate Data Dictionary

Relationships: Normalized data structures will be referenced by central design activities in the design of physical data structures to satisfy application level information requirements.

Approach: Throughout the data modeling process the group has been encouraged to look at the data independently of the processes, to capture the natural relationships of the data, and to assign attributes nonredundantly to their appropriate entities. As a result, the data model at this point will reflect an organization of data that is largely consistent with a normalized structure.

However, it is inescapable that the groups will view the data with some degree of process bias. Group members are selected because of an expertise in the function. This expertise is gathered through years of experience with the processes that have historically been executed to carry out the function. This experience will color the way the data is viewed and will influence how the group defines "natural" relationships.

The formal rules applied during the normalization process force adjustments to the data model which will remove most traces of process bias. These adjustments are critical to efficient operations in a modernized environment.

Although normalization is often commonly (and erroneously) regarded as a technical activity, it in fact depends on a thorough knowledge of the data and its characteristics from a functional perspective. The active participation of the group is critical.

The data model will be normalized at least to third normal form (and potentially to fifth normal form). Normalization to this level will ensure a correct, consistent, and nonredundant data model. Specifics of the normalization process will be addressed in formal training sessions.

During the normalization process additional entities, entity subtypes, relationships, and attributes may be identified. These will be documented as described in previous steps.

STEP 3.1.6 FUTURE DETAILED MODEL RECONCILIATION

Purpose: To reconcile the action diagrams and supporting information views with the detailed data model at the level of data attributes. To assure consistency between process and data information requirements and ensure completeness of the data model at the attribute level.

Description: In this step the attributes identified in the data model from Step 3.1.5 will be reconciled with the subprocess level information views (data elements) from Step 3.1.4 to ensure that these views can be satisfied to the element level. A process to attribute (CRUD) matrix will be developed using subprocesses (module-level data element requirements aggregated to the subprocess level) and the attributes from the entity descriptions. For purposes of this document data element and attribute are considered equivalent.

Outputs:

- o Subprocess to attribute (CRUD) matrix
- o Revised Action Diagram
- o Revised data model reflecting new attributes
- o Updates to the Corporate Data Dictionary

Relationships: This step is the end point for the detailed information system requirements definition activity. The information system requirements from this step will be prioritized in Step 3.1.8 and will become the basis for the information systems strategy. The outputs from this step will be among the documents provided to a central design activity tasked to develop future automated information systems.

Task 3.1.6.1 Aggregate Action Level to Subprocess Level Information Views

Purpose: To aggregate the detailed action level information views into subprocess level information views.

In Step 3.1.4 the action diagrams from Step 3.1.1 were partitioned into modules and data actions. An information view was developed for each process rectangle and defined in terms of data elements. The overall purpose of Step 3.1.6 is to ensure that the information requirements of the modules and data actions can be satisfied by the data model at the level of data elements. The specific purpose of this task is to consolidate the action level information requirements into a more manageable number by aggregating them to the subprocess level.

Outputs:

- o Subprocess level information views expressed in terms of data elements

Relationships: Aggregated subprocess level information views will be used to build the CRUD matrix in Task 3.1.6.2.

Approach: Action level data element requirements are aggregated to the subprocess level to eliminate redundancy and consolidate requirements. Given the large number of actions which will be required to represent a large system, creating an entry on a CRUD matrix at the level of the action would result in a document of unmanageable size. For this reason the information views will be aggregated back to the level of subprocess. A similar task was performed in Step 3.1.3 to aggregate action level information views to subprocesses. This task will proceed along the same lines and will result in aggregated views for the corresponding subprocesses. This is a refinement of the existing information views to reflect the more detailed data requirements portrayed in the action diagram. The resulting consolidated views will represent the sum of the data elements required to satisfy all of the actions within a given subprocess.

A simple way of collecting the data element information is by creating two working lists at the subprocess level of detail, one reflecting input data requirements and the other showing output data produced. These data elements are shown in the action diagram at the top right of the process rectangles for input, and the bottom right of the process rectangles for output. The lists should not duplicate the names of the data elements, even though they are used more than once by the actions or modules within a subprocess.

For example, assume that we identify twelve modules as part of three different process rectangles within a given subprocess. Each of these modules would have an information view expressed in terms of data elements. Any data element which was part of an information view for any of the modules would become part of the information view for the subprocess. The resulting subprocess information view will be defined in terms of data elements.

The identification of information views at the action level is essential to understanding the data requirements of the processes at the detailed level. The consolidation of these information views into a single subprocess level view is equally important in order to be able to deal with the inherent complexity at a reasonable level.

Task 3.1.6.2 Develop Subprocess to Attribute (CRUD) Matrix

Purpose: To develop a matrix that arrays all subprocesses against all of the attributes from the data model. This matrix is a refinement of the subprocess to entity matrix created in Task 3.1.3. It will allow for the reconciliation of the action-level data requirements (aggregated to the subprocess level) with the data attributes.

Outputs:

- o Subprocess to attribute (CRUD) matrix

Relationships: The subprocess to attribute matrix is a reconciliation tool which is the final level of reconciliation of the process model to data model. The matrix will be used to validate the availability of data elements in Step 3.1.6.3 and to validate the management of data elements in Step 3.1.6.4.

Approach: A matrix is created showing subprocesses on the vertical axis and data attributes on the horizontal axis. The nature of the relationships between subprocesses and attributes are then expressed in the context of the action(s) that the subprocess may take on each attribute. Potential actions are:

C Create: subprocess creates the data

R Read or Reference: subprocess references the data

U Update: subprocess modifies the data

D Delete: subprocess removes the data

For each intersection of a subprocess and an attribute on the matrix the group must consider potential action(s). If there is no action the intersection remains blank; otherwise, one or more characters are entered at the intersection as appropriate. The horizontal line extending to the right from each subprocess is one representation of the information view of that subprocess. The vertical line extending beneath each attribute represents management of that attribute within the function. Columns that contain only "R"s indicate that the attribute is probably owned by some other function and is part of the external interface of the function area being modeled. Since this document will be of considerable size and may require considerable manual effort, automated support is strongly recommended.

A sample CRUD matrix was included with Step 3.1.3. While the matrix from Step 3.1.3 arrayed subprocesses with entities, rather than with attributes, interpretation is similar.

Task 3.1.6.3 Verify Information Views With the Data Model

Purpose: To ensure that the consolidated action level information views identified can be satisfied by the data model.

Outputs:

- Revised data model
- Revised process model
- Updates to the Corporate Data Dictionary

Relationships: This task makes use of the information views developed by the process decomposition analysis (as revised in task 3.1.6.1) and the detailed CRUD matrix. Reconciled information views will become part of the final future process model.

Approach: In Step 3.1.4 the group developed an information view for each module or data action acting upon data. These information views were defined from a process perspective. They reflect the data needed for the associated action to execute its task. The subprocess level information views represent the data needed by all of the included actions within the subprocess to perform their tasks.

Reconciliation of views may be accomplished by examining the horizontal line extending from each process. If the CRUD indicators are consistent with the definition of the information view, then no action is necessary. Where they do not, corrective action may be necessary.

It is particularly at the attribute level that the problem may be one of semantics. This occurs where the process model refers to a data element by one name and the data model refers to the corresponding attribute by another name. In such cases the inconsistency should be isolated and the appropriate name used in both locations. Problems of this nature will be minimized where an effective data administration program is in force and where the Corporate Data Dictionary is used for both models.

It is also possible that data elements required by a subprocess may be "derived" data elements.

- Derived data elements are data elements whose values can be determined based on the values of other related data elements.

Where the data elements needed to derive the desired value are present in the data model, a problem does not exist. The appropriate definitions will be created in the data dictionary and the relationship among the derived element and the elements used to derive its values documented.

Also, the process model should be reviewed to determine the feasibility of modifying the process model to use different data elements to achieve the desired result. If this is readily done, the process model should be revised to reflect the new method.

Where inconsistencies are not resolved as described above, either the data model or the process model may need to be modified.

If it is determined that a required information view references an attribute that does not exist, the group will investigate to ensure that the subprocess, or any particular component actions and modules, belong within the future scope of the business function. Subprocesses, actions, or modules may be eliminated or moved to other functions as a result of this investigation. This may be a cross-function integration issue which must be coordinated with the affected work group.

If it is determined that the attribute is in fact required by the function, the data modelers will take appropriate action to add the attribute, complete with relationships and descriptions, to the data model.

Attributes which are not referenced by any subprocess should be investigated for potential missing processes or possible removal from the data model.

Since these same subprocesses were reconciled with entities in Step 3.1.3 it would be unusual to discover that a subprocess did not act on any attributes. If this should occur, the group will revisit the Step 3.1.3 Subprocess to Entity matrix and determine which entities the subprocess acted on. The group will then identify which attributes of that entity are of interest to the subprocess and ensure that these are included in the data model.

Task 3.1.6.4 Verify Attribute Management

Purpose: To ensure that all attributes are created, referenced, updated and deleted as appropriate within the function.

Outputs:

- Revised CRUD matrix
- Revised data model
- Revised process model

Relationships: The revised data and process models from this step will be packaged in Step 3.1.7 and will become the major part of the information systems requirements for the function.

Approach: The CRUD matrix will be examined to ensure that all attributes are managed appropriately. This is accomplished by examining the vertical columns extending beneath each attribute.

If there are no indicators in the column it is evidence that the attribute may not be needed by the function. This should be discussed by both the data and process modelers. The future scope should be considered in deciding whether management or reference of an attribute is appropriate for the business function. If a process has been overlooked, it should be fully documented and added to the process model and the CRUD matrix. If it is discovered that none of the subprocess (or their component actions and modules) reference the attribute, then the attribute should be removed from the data model.

The lack of Create indicators for an attribute indicates that the attribute may be imported from an external function. This should be verified through reference to the detailed action diagram and by contacting the external source of the data to ensure that the required data will be available to support the function when needed. This is a cross-function integration requirement. The group must ensure that a process to create the attribute either already exists or will be established in this or some other business function.

STEP 3.1.7 FUTURE FUNCTIONAL INFORMATION SYSTEM REQUIREMENTS

Purpose: To validate and document the future functional information systems requirements.

Description: The functional information systems requirements describe what an information system must do to support the needs of the future function. The requirements describe what will be accomplished, while the design performed by information systems professionals describes how the needs will be accomplished.

The group will validate the process and data models developed in Phase III. The group will then review the functional business plan from Step 2.1.9 to identify any additional policy, interface, and management requirements identified during the analysis. The process and data models and the additional policy, interface, and management requirements will be documented as the final list of functional information systems requirements.

Outputs:

- o Future functional information systems requirements
  - o Validated process and data models
  - o Validated subprocess to entity (CRUD) matrix
  - o Validated future scope, vision, and strategy
  - o List of additional policy, interface, and management requirements

Relationships: The functional information systems requirements will prioritized in the next step. The prioritized requirements are the basis for the information systems implementation strategy, the final output of this process.

Task 3.1.7.1 Validate Process and Data Models

Purpose: To package the process and data models as the list of information systems functional requirements.

Outputs:

- Validated process and data models
- Validated subprocess to entity (CRUD) matrix
- Validated future scope, vision, and strategy

Relationships: The information systems functional requirements are the primary input for the prioritized information systems requirements.

Approach: This task will be accomplished through four subtasks.

Subtask: Validate the Process Requirements from the Detailed Process Model

The group will list the subprocess documented during Step 3.1.4 in the order presented in the process to attribute matrix in Step 3.1.6. The group will compare the subprocesses with the scope of the future function, the vision elements, and the strategies. The group will identify any elements of the scope, vision elements, and strategies that are not represented by the subprocesses and will determine whether additional subprocesses should be identified to provide information systems support for the future vision. If additional subprocesses are required, the process model will be refined by the group. As part of this analysis, the scope, vision, and strategies will also be reviewed and updated if necessary. Following any changes, the group will examine the (revised) process model to ensure consistency of level, description or definition, and potential interfaces.

Subtask: Validate the Data Requirements from the Detailed Data Model

The group will list the entities developed during Step 3.1.5 in the order presented in the subprocess to attribute matrix in Step 3.1.6. The group will compare the entities with the scope of the future function, the vision elements, and the strategies. The group will identify any elements of the scope, vision elements, and strategies that are not represented by the entity-relationship diagrams and will determine whether additional entities or relationships should be identified to provide information systems support for the future vision. If additional entities or relationships are required, the data model will be refined by the group. As part of this analysis, the scope, vision, and strategies will also be reviewed and updated if necessary. Following any changes, the group will examine the (revised) data model to ensure consistency of level, description or definition, and potential interfaces.

Subtask: Validate the Integration of the Data and Process Models

The group will analyze the modules completed in Step 3.1.4 and the subprocess to entity matrix in Step 3.1.6 to identify any subprocesses that may be more effectively completed by another function. The strategy for assessing these functions is to perform a series of analytic steps.

The group will first identify all entities that are not created by subprocesses in this function. The group will review the entities that are updated and deleted to determine whether the subprocesses performed in this function manage these entities. If the operations are very limited and performed by very few subprocesses, then the possible "outliers" (subprocesses that may belong in a different function) should be listed for further review by the CIM functional integration teams.

The group will then review the entities that are read only to ensure that the action diagrams represent the entity as an interface. If the action diagrams describe a different type of subprocess, then the subprocess to entity matrix should be re-evaluated to ensure that the proper coding of the operation exists (create, update, or delete). If the operation is still read only, then the subprocess should be evaluated in terms of whether it provides direct information systems support of the vision elements. If the subprocess is key to the support, then the group will document it as part of the scope of the functional requirements. If the subprocess is not key, then it should be identified as a possible outlier to be performed by a different function. The potential "re-location" of the subprocess is then documented and provided to the integration team. If the subprocess defines the read operation as an interface, then the group will document it separately and provide it to the integration team for further analysis and placement in the relevant function.

There may be some cases where the "ownership" of the subprocesses is difficult to determine based on the re-analysis of the subprocess to entity matrix. In such cases, the subprocesses will be identified and provided to the functional integration team for review.

Subtask: Prepare Introductions to Validated Process and Data Models

The group will write a brief introduction to the validated process and function models. The introduction will include the use of the models, their relationship to the design process, and how they will be updated and maintained. Any unresolved questions of process ownership are also listed in the introduction. The introduction will not try to restate the descriptions of process and data requirements but may include some themes if appropriate.

Task 3.1.7.2 Compile Additional Policy, Interface, and Management Requirements

Purpose: To compile the list of additional policy, interface, and management requirements.

Outputs:

- List of additional policy, interface, and management requirements

Relationships: The additional requirements provide added information for the designers of the future information systems.

Approach: This task will be accomplished through three subtasks:

Subtask: Review the Functional Business Plan for Possible Additional Requirements

In this subtask the group reviews the business analysis in Phase II to identify any additional systems requirements they feel are not included in the data and process models. As part of the analysis of the functional concept in Step 2.1.5 the group identified requirements for operational and organizational changes that may need to be supported by the information system. In addition, technical and architectural needs, interface needs, and additional management needs may have been identified during the analysis and compilation of the functional business plan. Examples of these systems requirements may include security, performance, user interface, reporting, accuracy, workload, and training.

The group lists these textual requirements for analysis in the next subtask. This listing is meant to simply generate known system and performance requirements. It is not a substitute for the further analysis required by the design team once the requirements are provided as part of the implementation strategy, Step 3.1.9.

Subtask: Document Additional Requirements

The group will examine the list of textual requirements identified in the previous subtask and identify categories for classifying these requirements. They may include organizational, technical and architectural needs, interface needs, management needs, and any other particular requirements noted. The group then will list the requirements in these categories. The requirements are then complete and can be added to the documentation of the validated models prepared in Task 3.1.7.1.

Task 3.1.7.3 Prepare the Functional Information Systems Requirements Document

Purpose: To assemble the functional information systems requirements into a single document.

Outputs:

- o Future functional information systems requirements

Relationships: The functional information systems requirements will be used in the requirements prioritization in the next step.

Approach: The outputs from Tasks 3.1.7.1 and 3.1.7.2 will be compiled into a single document and then submitted to the CIM integration teams for review.

STEP 3.2.1 COMPOSITE HIGH-LEVEL PROCESS MODEL

Purpose: To define the Process Model through refinement of the processes and subprocesses identified in Step 2.2.4 into their component actions. Action diagramming, a new analysis tool to depict process models, will allow more detailed decomposition of the Business Function and serve as a basis for converting the Business Plan into information system requirements. The initial action diagram will be progressively refined in the following tasks and subtasks.

Description: In this step the group will further decompose the previously described processes and subprocesses into their component actions. <u>Processes (and possibly subprocesses) identified in Phase II may need even further refinement before it is appropriate to attempt identifying the detailed actions.</u> The decomposition methodology creates a continuum of more and more detailed descriptions of the processing requirement. Different Functional Groups will reach different levels of processing when discussing processes or subprocesses.

- Actions are the lowest level of activity within the function. They are the detailed logical constructs required to perform a function.

A training session will be required to familiarize the group with action diagramming techniques. At this level, the graphic tool developed is called a Macro-Level Action Diagram and represents an abstract view of the process aspects of the Business Function.

A one sentence description of each action will be prepared. These action definitions, as incorporated into the macro-level action diagram, provide a detailed view of the function and serve as the basis for the final level of data handling analysis performed in Task 3.4.1.2.

As the processes and subprocesses are decomposed to the action level, a more specific perspective on their data requirements will become apparent. This perspective, called an information view, will be documented and used with the data model during the reconciliation process.

- An information view is a collection of data (inputs or outputs) required by a process to complete its processing activity.

The information view provides a more detailed description of the information classes as defined in Task 2.2.1.3. An information class is composed of two or more logically related information views.

Outputs:

- Revalidated composite functional flow diagram
- Macro-level action diagram
- List of action definitions
- List of information requirements (data elements) and definitions

Relationships: The macro-level action diagram will be further refined in Step 3.2.4 to reflect input, output, and data handling action details. The final action diagram will be used for evaluating the future and composite requirements during the prioritization and implementation steps.

Task 3.2.1.1 Create Macro-Level Action Diagram

Purpose: To define the process to action relationship and to show the information flow to the action level by taking the Phase II defined processes and subprocesses and structuring them in an action diagram format.

Outputs:

- o Revalidated composite functional flow diagram
- o Macro-level action diagram

Relationships: The macro-level action diagram is based on the functional flow diagram produced in Task 2.2.4.2 and provides information for Tasks 3.2.1.2 and 3.2.1.3. The conversion from flow diagramming to action diagramming is required to allow introduction of more specific process details (action and logic) and to provide a sound foundation for the system designers.

Approach: In a hierarchical manner the functional flow diagram from Task 2.2.4.2 will be converted to an action diagram using a subset of the action diagramming symbols. Before beginning the action diagram, the composite functional flow diagram developed in Task 2.2.5.1 should be reviewed to confirm that the functional flow diagram is accurate and to assure that the comprehensive view of the function is well understood prior to diagramming. The action diagram will be expanded later in Phase III to reflect input and output data handling actions. The first step will only use a subset of the diagramming symbols to prepare the high-level view of the function. This diagram will be further enhanced as the decomposition process proceeds.

The action diagramming methodology is comprised of approximately 20 graphic symbols. During this task these high-level processing control symbols will be used to convert the functional flow diagram to a high-level action diagram. Figure 1, below, graphically depicts the symbols used in the action diagramming methodology and should be referenced when reading the symbol descriptions. The following narrative expands upon the graphic representations in Figure 1.

Process Control Symbol	Description
Title	The title is the name of the function, functional activity, process, or subprocess. (See item 1 in Figure 1.) It is a processing section identifier and can serve as a qualifier for duplicative sections. Titles always begin with an asterisk. This serves to highlight and differentiate titles from the other diagramming symbols.

Titles are used in high-level discussions of the functional activities, processes, and subprocesses involved in a function. A title is usually followed by a series of actions and logic elements which specify the activities within the functional activities, process, or subprocess.

Process Rectangle

This four sided figure has several key components for defining the function processing. (See item 2 in Figure 1.) It begins with and contains Title(s). In a top-down sequence, it contains the name of the processing steps.

Outside of the rectangle on the top right are the process inputs (data that the process requires) and outside on the bottom right are the process outputs (data that the process produces). The inputs and outputs are information views, reports, products, and interprocess communications.

This identification of inputs and outputs allows for verification that every functional activity input goes into some process and that every output comes from a process. In this way, the high-level data requirements identified for each function can be traced to a decomposed process, thereby assuring that a complete decomposition has been achieved.

Initially the action diagram will be simply a graphic containing a series of Titles and Process Rectangles identifying input and output to the processes. This simple block structure must directly relate to the Data Flow Diagram from Step 2.2.4. Through the addition of the following diagramming symbols, this fundamental action diagram will be successively decomposed into a detailed representation of the processing activities within the function.

Brackets

Brackets are used to distinguish a series of processing steps independent of input or output considerations. (See item 3 in Figure 1.) Typically they will be used within a process rectangle to highlight a series of steps which are associated in some logical fashion and are at the same level of detail.

Brackets act very similarly to Process Rectangles. In fact, Brackets can be considered a short-form of Process Rectangles. They differ in that Brackets emphasize a set of actions and denote that the actions are logically related in accomplishing the processing requirement being decomposed.

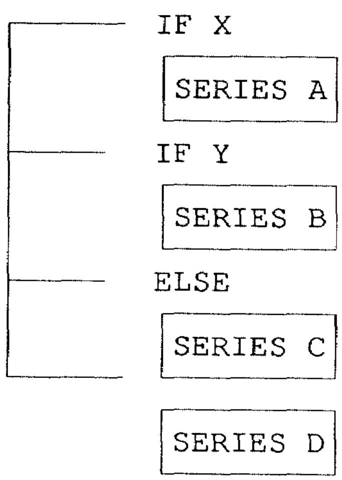
When Brackets are located inside of other Brackets, it is referred to as Nesting. This is a frequent occurrence as the Action Diagram proceeds to more detailed levels of decomposition.

IF

The IF symbol represents conditional performance of a sequence of activities. The IF symbol can appear at the top of a Process Rectangle and portray the conditions under which the processing activities occur. Typically IF symbols will be used with brackets to identify a series of processing steps subject to the same logical condition. IF symbols can contain an ELSE clause which represents activity execution when none of the IF conditions have been met.

IF symbols allow the representation of the conditions under which the following series of processing actions should be performed. That is, IF this is true, then do the following. The optional ELSE clause provides for a default processing sequence if none of the IF conditions are met.

In the example presented below, IF X is true then perform action series A. Dropping down a level, IF X is false and IF Y is true, then perform series B. Dropping one further level, IF X is false and Y is false--then perform series C identified under ELSE. After performing the "true" processing series, control flows to the bottom of the IF symbol, exits, and performs the next sequential action(s) (series D in the example).



When combined with the ELSE feature, IF symbols have a true and a false section. When the IF condition is true, only the statements following the IF condition are performed. When the IF condition is false, only the statements following the ELSE feature (if it is present) are performed. This logic can get fairly tricky if compound IF conditions (connected by OR or AND) are used. For example, "IF X AND IF Y OR IF Z" can be confusing as a condition. To alleviate some of this confusion, use simple IF symbols whenever possible. Compound IFs can often be broken down into a series of simple IFs. As a last resort, parenthesis can be used to try and clarify the meaning of a tricky compound IF. Thus, the original example could be restated with parenthesis as "(IF X AND IF Y) OR IF Z".

Nesting

Nesting represents a processing activity which contains one or more lower level processing activities. (See item 6 in Figure 1.) It is depicted by a smaller Process Rectangle completely enclosed within a larger Process Rectangle. All of the basic rectangle features apply to a Nested Rectangle. As noted previously, Nesting will most often occur at the Bracket level.

When Nesting symbols are used in combination with IF symbols, the results can get a little tricky. Particular care should be taken when combining IF symbols and Nesting to assure that the process flow described in the diagram is the one required by the function.

Repetition

Repetition is the process of repeating an activity or a section of activities several times. (See item 4 in Figure 1.) This is represented by a double bar at the <u>beginning</u> and <u>end</u> of the repetition sequence. Repetition control structures or qualifiers can be identified at the top of the sequence, much like a Title. Such qualifiers can include DO WHILE, LOOP WHILE, FOR, REPEAT, and others. Since the analysis is language independent, all that is important is consistency in applying the repetition nomenclature.

If Repetition is viewed simply as a way to avoid having to duplicate a processing description, its use remains fairly straightforward. However, instances of Repetition, particularly when combined with IF/ELSE symbols and Nesting can significantly increase the complexity of the process descriptions. As the objective of Phase

III is analysis and not design, use of Repetition should be logically driven by the process requirements and not used as a way of representing "how" the processing should be performed.

EXIT

The EXIT symbol represents termination of the activity sequence and rescinding control to another activity sequence. (See item 5 in Figure 1.) EXIT can be used as an end of processing placeholder (the bottom) whereby control is returned to the next higher level of processing. Alternatively, EXIT with an arrow attached can represent an immediate branching from this processing level to another processing level. In this role it could be considered a form of GOTO. EXITs can be conditioned by using IF or Repetition control structures, as in IF X=Y EXIT.

An EXIT is simply a form of escape which allows redirecting the flow of processing. Often it is used to end a repetitive series of processing activities to return to a higher level of processing. Although the EXIT symbol is an easy way to jump out of a Nested series of Brackets or Process Rectangles, it seriously diverts the flow of processing from a sequential top-down flow. For this reason, EXITs should be used sparingly and only when they simplify the logic. When an EXIT seems necessary, examine the IF symbol conditions and see if they can be modified to eliminate the need for an abrupt EXIT.

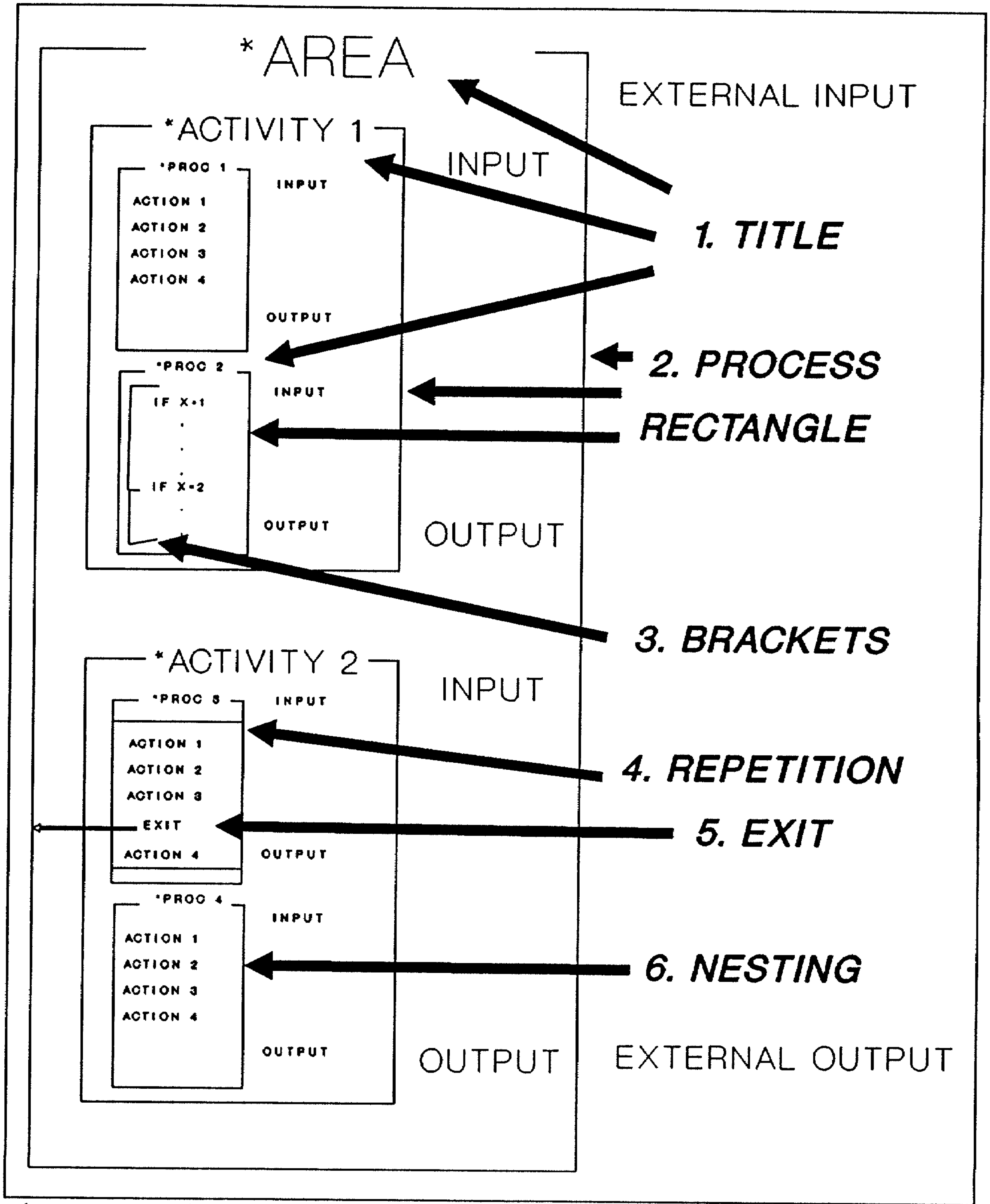


Figure 1 Action Diagramming Tools

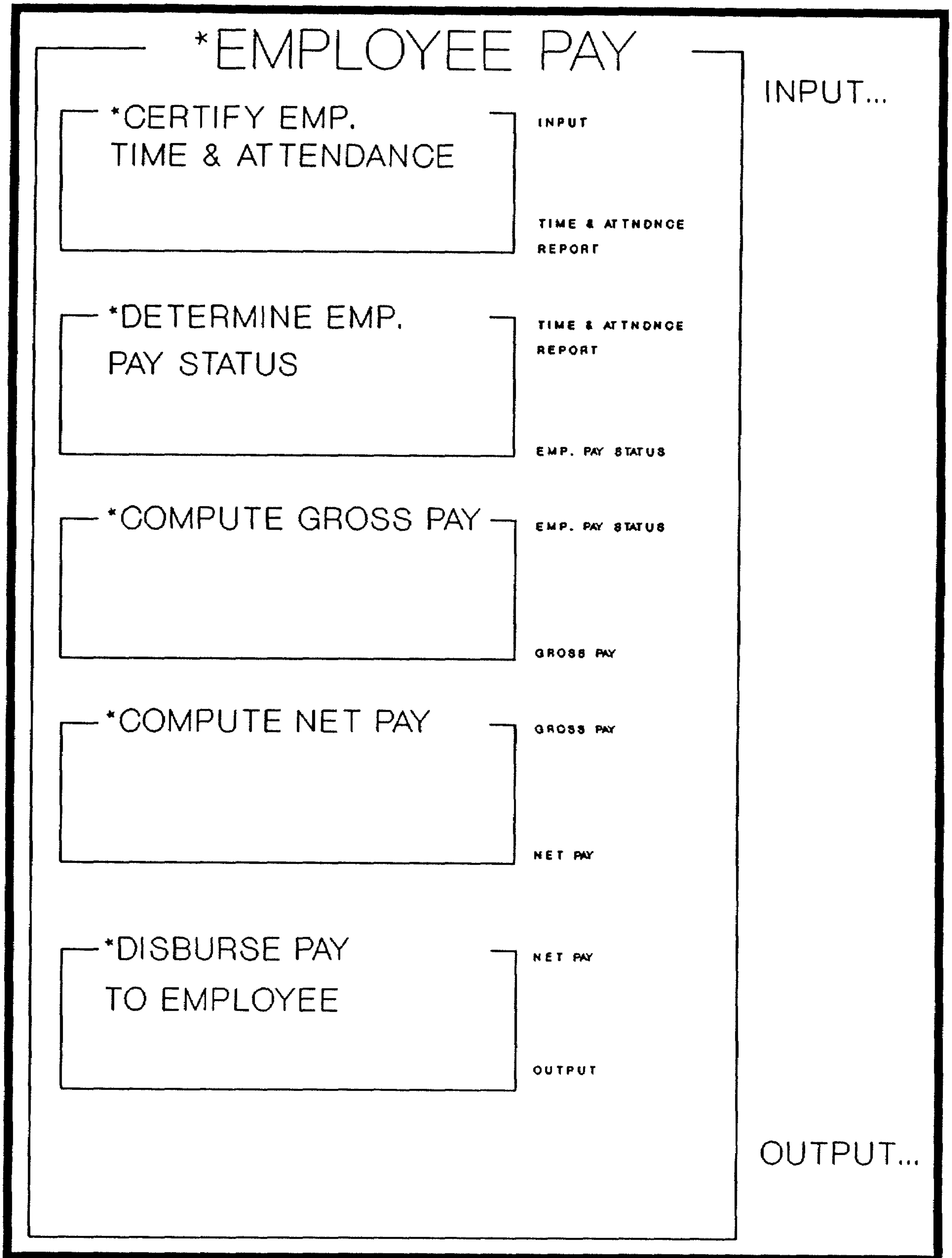


Figure 2 Action Diagram of the Step 2.1.8.1 Flow Diagram

Subtask: Review Composite Functional Concept and Confirm the Accuracy of the Flow Diagram

The functional flow diagram produced in Task 2.2.4.2 should be reviewed to verify that the Function is clearly and accurately represented in the diagram. The flow diagram must be intensively reviewed to assure accuracy, as it provides the "big picture" view of the function and serves as the point of departure for the detailed process decomposition. Processing bubbles identified in the flow diagram should accurately represent the Function and be established in a fashion which will permit further decomposition. Within a function, processing area overlap and duplicative data handling responsibilities should be considered for elimination. Major data flows should be identified reflecting their associated (source and destination) process bubbles and relationships. The output of this subtask should be a final version of the flow diagram which rigorously presents the processing and data used in the Function.

Subtask: Convert Processes and Subprocesses to Actions and Identify Input and Output

This subtask is begun by preparing a large processing rectangle to contain all of the functional activities, processes and subprocesses defined previously in Phase II. This rectangle should be titled with the name of the function being analyzed. It is critical that all external inputs and outputs to the function be listed in the appropriate locations outside of this box. These external information views represent this function's linkage with other functions.

- If information used within a function is obtained or given to a process outside of this function, then that information view must be listed in the appropriate location (<u>input at the top right, output at the bottom right</u>) outside of the largest functional processing rectangle. These entries represent this function's external interfaces.

Through direct reference to the functional flow diagram produced in Task 2.2.4.2, process rectangles should be constructed which reflect the sequential and hierarchical structure used in the flow diagram.

Process and subprocess input and output requirements should be entered in the appropriate areas of the action diagram in a manner which reflects a "trickle down/float up" structure. This should make apparent where inputs and outputs to the processes and subprocesses originate and their flow from process to process should be evident. There is no distinction in the diagram between process to process communication and process to data store to process communication.

Figure 2 illustrates the format and level of detail required for this subtask. This example is an action diagram of the flow diagram presented in Task 2.1.8.1. The Employee Pay model contains the series of processes and subprocesses identified in the processing rectangles. These processing rectangles will be further developed in the next subtask to identify the actions required to accomplish the processing. In the example not all of the input and output requirements (information views) were provided. However, the input and output information views are integral to the decomposition process and should be analyzed with an intensity equal to that of the process decomposition.

The information views (and their data subcomponents) are critical for comparison with the data model during later steps in Phase III. Note that for external data, only the information views (and data sources) are documented, not the processes that produce the data. This source and destination documentation is primarily used as an aid to the design phase and to identify opportunities for integration across functions.

Subtask: Define Processing Control Structures

The conditional control logic required by some of the processes is identified in this step. The addition of process control structures clarifies the order of processing and identifies any conditions which influence the order of processing. Principally this involves the identification of the actions which comprise a process (what processing activities need to be performed) and the definition of IF structures and IF/ELSE conditions.

Potentially these structures can cause the repetition of some processes or subprocesses in order to clarify the control logic. This duplication need not be avoided. However, once the IF control structures are defined, the overall diagram should be reviewed to determine if it can be simplified by reordering or restructuring some of the processing activity sequences. Complex IF symbols should be reworked to simplify them as much as possible. This may require breaking them into multiple IF structures. This restructuring should only make use of Nesting and EXIT symbols to represent the processing logic when their use simplifies or clarifies the flow at processing.

As the decomposition process proceeds to the action level of detail, it will be apparent that the action diagram has expanded beyond a single page. This is the transition from a macro-level action diagram to a standard action diagram.

If an automated symbol is being used, there should be no difficulty in taking a modular approach to the diagramming activity. The initial process level of detail (as performed in the previous subtask) should be retained on as few physical and logical pages as possible. This will help in gaining an overall perspective of the function much as was attained in the original functional flow diagram. However, as the processes and subprocesses are decomposed into actions and control logic, they will probably require the use of more than one page.

Consistency across diagrams and pages regarding control structures and nesting levels is critical. Due to the number of details and actions which need to be diagrammed, as well as the normal iterations occurring in any analytical activity, an automated diagramming symbol is recommended.

Subtask: Cross-Function Integration

It is not unusual for a functional group to find they have identified processing requirements which should be the responsibility of another function. For example, the Contract Payment group (or possibly the Materiel Management group) might identify a processing requirement to maintain a vendor or contractor list. In fact, this process is probably best performed by the Contract Management group who intuitively would have primary responsibility for the list of contractors. To avoid redundancy and confusion, these duplicative areas should be coordinated with the affected functional groups and a determination made as to assigned responsibilities (many users, one owner). In coordination with the other functional groups, a structured walk-through of the action diagram should be presented. This will allow the group to review its diagram for completeness and quality of concept. A briefing should be prepared as a discussion vehicle, and in coordination with the other functional group, a meeting should be held to identify and delineate areas of responsibility.

Task 3.2.1.2 Develop Composite Actions Definitions

Purpose: To further refine the action level process decomposition by defining all of the identified actions.

Outputs:

- List of action definitions

Relationships: The list of actions is obtained from the macro-level action diagram created in Task 3.2.1.1 and is used in the reconciliation process in Step 3.2.3.

Approach: Obtain a list of action names from the macro-level action diagram and explain their meaning using one sentence narratives. Typically this will include all of the lines which do not have either a title or processing control symbol.

The action names should be qualified by process or subprocess for further clarification and a list prepared compatible with the existing process definitions (see Task 2.2.2.1). Higher order actions, which are groups of more primitive actions, should be broken down to the lowest action level prior to definition. Actions which cannot be clearly defined in a one sentence definition should be considered for further decomposition. It is not unusual for higher order actions to masquerade as primitive actions to this point in the analysis. Through the diagramming and definition processes the work group should resolve the actions to their primitive level to complete this step.

An action can be considered adequately resolved when its purpose can be described in one sentence. For example, in the Contract Payment Function "authorize payment of a contract" could be an action definition. The action and definition should address the "what" aspect of the function and must not be influenced by the "how" design aspect.

Task 3.2.1.3 Define Compoiste Action Information Requirements

Purpose: To identify the information requirements (data elements) of the function at the action level. Detailed data specifications which have been casually identified during the decomposition process will be collected and defined for later use during the reconciliation with the data model.

Outputs:

- o List of information requirements (data elements) and definitions

Relationships: This list is obtained from the macro-level action diagram created in Task 3.2.1.1 and is used in the reconciliation process in Step 3.2.6.

Approach: The input and output requirements (information views) identified in the action diagram will be used as reference points for identifying some of the data element requirements of the processing actions.

- o Data elements are elementary types of data which collectively form a processing information view. Data elements relate directly to the data attributes identified in the data modeling process. For example, when viewed from a processing perspective the information view called employee may be composed of the service identifier, social security number, age, sex, and marital status data elements (among others).

This task serves two major functions. It encourages clarification of the information views required in the 3.2.3 Composite High-Level Model Reconciliation step and begins the collection of data elements required for the 3.2.6 Composite Detailed Model Integration step.

Typically the action diagrams identify information views at the process level and above. At the action level it is not unusual to identify major data elements as well as information views. In this task any data elements which have been recognized while identifying the information views will be organized into a list of definitions. <u>This should not be an intensive dissection of the information views, but rather the collection of incidentally or already apparent data elements.</u>

Using the input and output sections of the action diagram as a starting point, examine the actions to collect any data elements which have already been identified and to consider what data elements are required to perform the processing. For this analysis the actions should be viewed as processors which transform the input into the output. All data elements identified as needed to perform this transformation should be listed and defined in a short narrative form. This narrative should specify any apparent special roles (key, index, etc.) which this data element performs for the information view.

STEP 3.2.2 COMPOSITE HIGH-LEVEL DATA MODEL

Purpose: To continue describing at a high-level the data portion of the composite information requirements for the function.

Description: In Phase II the Composite Functional Information Model was created. In this step the group will continue to use Entity-Relationship diagramming techniques to build upon this model and to create the initial high-level Composite_Data Model (high-level in the sense that not all entities and entity subtypes may be identified and that it will not be fully attributed). This model will be compared with the set of action diagrams developed in Step 3.2.1 to reconcile specific information processing requirements and to maintain consistency between the data and process perspectives of the function.

In the Composite Functional Information Model from Step 2.2.3 only major entities were identified. In this step the group will identify all other entities and will begin to identify entity subtypes. Candidate identifier attributes and other significant attributes identified during the data modeling process will be documented in the standard Corporate Data Dictionary. Relationships among entities and entity subtypes will be captured. An entity-relationship diagram will be drawn.

Integration of entity descriptions, entity subtype structures, and attribute definitions with other business functions and with other analytic paths (i.e. the future functional path and the current information system paths) will occur in the final task of this step.

This step will require the assistance of a trained data modeler.

Outputs:

- High-level data model consisting of
  - Entities and descriptions
  - Entity subtypes and descriptions
  - Entity and entity subtype relationships
  - Relationship descriptions
  - Entity/entity subtype structures
  - Entity relationship diagram
  - Business rules
- Candidate identifier attributes

- o Attribute names and definitions
- o Updates to the Corporate Data Dictionary

Relationships: In Step 3.2.3 the data model will be reconciled with the action diagram (process model) from Step 3.2.1 to ensure that the processes required to effectively manage data entities have been identified and to determine whether the information requirements (action diagram information views) necessary for the successful execution of processes can be satisfied.

The entity descriptions, entity subtype structures, and attribute definitions identified in this step will be integrated with similar products for other business functions to promote corporate wide data integrity.

Task 3.2.2.1 Identify Entities and Entity Subtypes and Document Their Descriptions

Purpose: To identify entities and entity subtypes necessary to support the function and to develop descriptions of these entities and entity subtypes.

Outputs:

- Entities and entity subtypes
- Entity and entity subtype descriptions

Relationships: The entities and entity subtypes with their descriptions will be used to build the entity relationship diagram in Task 3.2.2.6.

Approach: This task will be accomplished through three subtasks.

Subtask: Identify Entities

- A data entity is something of lasting interest which is uniquely identifiable and about which data must be stored. An entity can be tangible such as a person, place, or thing; or intangible such as an event or concept.

In this step the group is not building an entirely new data model, but rather is continuing to build on the model begun in Phase II. With the high-level entities from the Composite Functional Information Model in Phase II as a reference, the group will identify additional entities necessary to execute the function.

In Phase II the group was encouraged to identify only major high-level entities. In this task the group will consider all entities and entity subtypes necessary to support the function, including all entities needed to support the individual processes and subprocesses identified in the High Level Action Diagram (Process Model) from Step 3.2.1. To ensure that information linkages to other functions are developed in this task, both the entities that lie within the function and those that are closely related but lie outside the function must be identified. The latter type of entity is best identified by analyzing the descriptions of external interfaces defined in Task 2.2.2.4.

The same series of tests used in Task 2.2.3.1 to identify entities for the Composite Functional Information Model may be applied to determine if candidates are in fact entities:

- Is it necessary to collect information about the candidate in order to manage or execute the function? If yes, what kind of information?

- o Does it have meaningful relationships with other potential entities?

- o Is it distinguishable from other potential entities?

- o Can a single occurrence of the potential entity be uniquely identified?

The entities identified in this task, together with those from Phase II, will become the basis for the entity-relationship diagram to be developed in Task 3.2.2.6.

Subtask: Identify Entity Subtypes

While developing the Composite Functional Information Model in Phase II the group was specifically directed not to identify entity subtypes. In this step the group will identify subtypes, as appropriate. In general, this step should identify subtypes to the first level below the entity.

- o Entity subtypes are a hierarchical decomposition of an entity based on some criteria. They are subsets of an entity established to record information specific to the subset and which have distinct associations to other entities. Ideally subtypes fully partition the entity (every instance of an entity will belong to one of the subtypes) and there will be no overlap (each instance of an entity will belong to only one subtype).

For example, Individuals may be either Civilian Employees, Military Members, or Customers. It may not be sufficient to identify the single entity, Individual, and accurately model the function's data. The function may treat Civilian Employees differently than Military Members and Customers. Since each of these are categories of the entity type Individual, they are all subtypes of the Individual entity. Individual, in turn is considered a supertype of the entities Civilian Employee, Military Member, and Customer. A relationship originally expressed as "Individual places Customer Order" could be more meaningfully described as "Customer places Customer Order". This can only be done if Customer is identified as an entity subtype.

There are generally multiple ways to partition an entity into entity subtypes. Individual could be subtyped as male or female, active or retired, or by race just as easily as we partitioned by Civilian Employee, Military Member and Customer. The way the data is partitioned into subtypes changes the way the data is viewed. The method of subtyping selected should be that which most meaningfully represents the data as viewed by the business function. This will be determined by the data steward (described in Task 3.2.2.7) for the entity.

The following logic can be used to determine if an entity subtype exists:

Entity B is a subtype of entity A if

- entity B and entity A represent the same object in the real world; and
- entity B has all of the attributes (properties) of entity A plus some additional attributes of its own (This will be clarified in a later step when entities are fully attributed.); and
- for every occurrence of entity B there exists precisely one occurrence of entity A while the reverse is not necessarily true, i.e., there may not be an occurrence of B for every A.

Even where these criteria are met, entity subtypes should only be identified where they serve some clear business purpose and not merely as an academic exercise. Upon close examination virtually all entities can be categorized into subtypes. The question must be asked whether the entity subtype is treated differently (has different relationships to other entities) in the business than the entity supertype. If the subtypes are not treated differently, there is no reason to formally document them.

Conceptually, there may be multiple levels of entity subtypes. Military Members (a subtype of Individual) may be categorized as Officer and Enlisted. Officer (a subtype of Military Member) may be Active, Reserve, or Retired. It is possible to spend considerable time exploring the levels of entity subtypes and the resulting data structures.

The purpose of this step, however, is not to explore the data to its lowest level. This will be accomplished in a later step. Rather, the group is trying to establish a data model which will clearly represent the information requirements of the functions and to do so as expeditiously as possible. It is anticipated that this can be accomplished by structuring the data model to the first level of entity subtypes. If further decomposition is clearly called for to establish meaningful relationships, then it is certainly acceptable to identify those subtypes now. Additional decomposition may also be accomplished in Step 3.2.3 as the information views necessary to satisfy specific information requirements at the action level are reconciled.

A preliminary activity to the identification of entity subtypes should be the identification of any undiscovered entity supertypes. It is possible that in the original identification of entities some number of entities were identified which were actually subtypes of the same supertype. For example, we may have identified Civilian Employee and Military Member as entities without having recognized that they are both subtypes of the same supertype - Individual. While it is our intent to ultimately identify these subtypes, the supertypes must be established first if subsequent modeling is to proceed in an orderly fashion. In general, entities are of the same supertype if they have common identifying attributes and do not have clearly distinct business meanings.

Subtask: Describe Entities and Entity Subtypes

In this step the group will capture descriptive information about entities and entity subtypes, including a definition and a statement of the business purpose for each entity or entity subtype described.

Each entity must be uniquely and unambiguously defined so that potential users who are not involved in the analysis effort will be able to determine exactly what is included in, and excluded from, the entity. A precise definition of each entity is essential. Ambiguities may lead to redundant and inconsistent implementation. Definitions will focus on what the entity is and not how it is used.

Entities identified during Phase II were described at that time. The group has since acquired a more thorough understanding of the function and of its data. The Phase II descriptions will be reviewed in light of this increased understanding and modified as appropriate.

Once entities are described, entity subtype descriptions must also be developed. Each entity subtype must be described in a manner which makes clear the criteria used to distinguish it from the entity supertype and from all other entity subtypes at the same level.

Task 3.2.2.2 Identify Candidate Identifier Attributes

Purpose: To identify candidate attributes which uniquely identify each instance of an entity. To record the definitions of these attributes in the Corporate Data Dictionary.

Outputs:

- Candidate identifier Attributes
- Updates to the Corporate Data Dictionary

Relationships: In ensuing tasks the group will identify relationships among entities and will describe the characteristics of those relationships. The ability to accurately and completely identify relationships depends on the level of understanding of the entities in the model. Documentation of candidate identifiers will enhance this understanding and facilitate identification of relationships in Step 3.2.2.4.

Candidate identifiers will become part of the composite information requirement.

Approach: For each entity or entity subtype one or more candidate identifiers will be documented.

- Attributes are characteristics of data entities which are describable in terms of some value. They are the lowest level of information about data. For purposes of the Phase III Process Guide, attributes are synonymous with data elements.

A candidate identifier is an attribute of an entity which uniquely identifies a given instance of the entity from all other instances of that same entity. Name, Social Security Account Number, Employee ID, Address, and Phone Number might all be attributes (or properties) of the entity Employee. In this case Social Security Account Number, Name, and Employee Id would all be candidate identifying attributes which might uniquely identify each instance of the entity Employee.

Sometimes multiple attributes are necessary to uniquely identify an entity. Both the Flight Number and the Date might be necessary to uniquely identify each occurrence of the entity Airline Flight. In this case a concatenated, or compound, identifier can be established consisting of multiple attributes.

The identifier of an entity subtype is generally the same as the identifier of the supertype. Sometimes, however, additional attributes are necessary to fully qualify instances of a subtype. In this case the additional attributes should be concatenated and the resultant compound identifier used as the identifier for the subtype. In other cases completely different candidate identifiers may exist. Customers and Employees might both be subtypes of Individual. While Social Security Account Number would seem like a strong candidate identifier for Individuals, we might not capture this information about either the subtype Employee or the subtype Customer. A candidate identifier for Customer would be Customer Account Number, while for Employee we could use the Employee Account Number.

While it is tempting to focus on a single identifier at this point, caution should be exercised. The best identifier may be different for different functions. Selection of the identifier may be deferred until the implementation of physical data structures.

As candidate identifying attributes are identified they will be named in accordance with corporate standard naming conventions documented by the corporate data administrator. Final approval of all attribute names must be by the corporate data administrator and must not be accomplished by individual group members. This control over attribute naming is key to preserving data integrity and is a first step in integrating data across functions.

Task 3.2.2.3 Document Additional Attributes

Purpose: To collect information about those additional attributes (beyond the candidate identifier attributes identified above) which the group feels are necessary to document at this point.

Outputs:

- Attribute names and definitions
- Updates to the Corporate Data Dictionary

Relationships: Attribute definitions will be entered in the data dictionary for reference by functional groups, central design activities, and other interested parties throughout the life cycle of the business function.

Approach: It is not the purpose of this task to fully attribute data entities. Rather, the identification of attributes beyond the candidate identifiers should be deferred until Step 3.2.5 whenever possible. This task recognizes that during the modeling process common attributes are often identified which the group may feel are significant and worthy of immediate documentation.

New attributes will be named (in accordance with the standard naming conventions mentioned above), defined, and related to the appropriate entity or entity subtype. Entries for all new attributes will be made in the Corporate Data Dictionary under the control of the corporate data administrator.

As attributes are identified, the existing Corporate Data Dictionary will be researched to determine if the attribute already exists and if the definition as written is applicable. Problems with existing data names and definitions are data administration issues which must be resolved under the authority of the corporate data administrator and consistent with the concept of data ownership discussed in Step 3.2.2.7.

The principle of inheritance will be applied in the identification of attributes.

- The principle of inheritance allows lower level subtypes within a hierarchy to inherit attributes from the higher level. An attribute will be included at a higher level only if it applies to all of the subtypes. An attribute which only applies to some of the subtypes will be described at the level of the subtype. For example, all mammals are warm blooded. The attribute "warm blooded" would apply at the level of mammals. Only some mammals walk on four legs. The attribute "walks on four legs" would not apply at the level of mammal, but rather would apply individually to each of the mammal subtypes which walk on four legs (giraffes and horses, but not apes and man).

In accordance with the principle of inheritance, attributes will be related to the highest appropriate level within an entity/entity subtype hierarchy. Consider our categorization of the entity Individual into the entity subtypes Civilian Employee, Military Member, and Customer. Since the attribute Name applies to all subtypes (Civilian Employee, Military Member, and Customer) of the supertype Individual, the attribute should be related to Individual and will be "inherited" by the subtypes. It therefore becomes unnecessary to relate the Name attribute to each of the subtypes individually. The attribute describing "customer account balance" would meaningfully apply only to Customer and would therefore be captured as an attribute at the level of the subtype, Customer.

The amount of attribute information collected at this point may be minimized. Collection of detailed information about default values, optionality, security, and the like may be deferred until Step 3.2.5.

TASK 3.2.2.4 Identify Relationships Among Entities and Entity Subtypes

Purpose: To identify relationships among entities and entity subtypes.

Outputs:

- Entity and entity subtype relationships
- Relationship descriptions
- Entity/entity subtype structures

Relationships: Entity relationships will be used to build the entity relationship diagrams (ERDs) in Task 3.2.2.6.

Approach: A relationship is an association among entities or entity subtypes. Relationships have properties that describe what they are and how they operate.

Relationships among entities and entity subtypes are expressed in the form of a verb or verb phrase. The relationship between the entity Students and the entity Classes is that Students "attend" Classes and Classes "are attended by" Students. This is the most common type of relationship on an ERD and is the type that was introduced in Step 2.2.3.

There is, however, a second type of entity relationship. This is the relationship that exists between levels within an entity/entity subtype hierarchy. That is, the relationship between a high level entity (entity supertype) and what may be called its composite entities (entity subtypes).

The distinction between an entity and its entity subtypes exists as a result of the categorization of an entity into subtypes based on some property or attribute. This concept works much the same as the way animals are categorized based on whether they are cold or warm blooded, whether or not they eat meat, or whether they have two, four, or more than four legs. In each case a single attribute may be used to categorize into subtypes. The relationship between the higher level entity and its subtypes is expressed in terms of the attribute which makes each occurrence a member of a specific subtype, rather than in terms of some verb. In our animal example, we might use attributes such as "method of body temperature regulation", "diet", or "number of feet" to categorize the animal entity into subtypes. Doctor, Nurse, and Technician are related to the supertype Employee by the attribute "occupation" and this relationship is expressed by the attribute "occupation" rather than as a verb. (Note: In a payroll function this distinction might not be sufficient to justify partitioning into subtypes. In a hospital function such partitioning would probably be appropriate, as Doctors, Nurses, and Technicians all behave differently in the hospital business. The decision whether to categorize into subtypes will be based on the groups understanding of the function.)

For purposes of definition, we will refer to this second type of relationship as a "subtype relationship." All other relationships will simply be referred to as "relationship". This document will make few references to subtype relationships.

Where difficulty is encountered in expressing a relationship, the situation should be examined closely. It may be that multiple relationships exist between two entities and should be expressed as such. It may be that the true relationship is with an as yet unidentified entity subtype, in which case that entity subtype should be identified. It might also be that the relationship is a complex relationship.

Thus far we have talked exclusively about binary relationships, or simple relationships between two entities. There are also complex relationships. These are relationship that exist among three or more entities. They are sometimes referred to as "n-ary" relationships. The relationship among a doctor, a patient and a surgical procedure can be expressed as three binary relationships (doctor to patient, patient to surgical procedure, and doctor to surgical procedure). This same relationship can be more meaningfully expressed as a complex (ternary, or n-ary) relationship. Specific techniques for addressing complex relationships will be addressed in formal training.

Relationships between entities should be established at the highest appropriate level. The "is employed by" relationship from Organization should be made to the supertype Employee rather than to each of the subtypes Doctor, Nurse, and Technician. The "provides primary care" relationship from Patient, however, should be established directly to the subtype Doctor.

As relationships are identified they will generally be between entities that clearly belong within the scope of the function being documented. Some relationships may involve an entity which just as clearly belongs in another business function. Such relationships should be documented and the entities in question represented on the entity relationship diagram developed in Task 3.2.2.6. However, rather than create descriptions of those entities which belong within the scope of other functions, the descriptions developed by those functions can be used. Resolution of potential inconsistencies in this area will be accomplished as a natural part of the ongoing integration process.

All relationships should be documented with a narrative that meaningfully describes the relationship.

TASK 3.2.2.5 Document Business Rules

Purpose: To document business rules identified in the data model. Business rules may result from the constraints placed on relationships by business practices.

Outputs:

- o Business rules

Relationships: Reference will be made to the business practices identified in Task 2.2.2.2. Business rules will become part of the composite information requirements for the function.

Approach: Business rules describe characteristics of the entities and their relationships that reflect the business practices of the function. Business rules may apply either to the entities themselves or to the relationships among entities. Application of business rules preserves the integrity of the data.

The constraints placed on relationships due to business practices identify one category of business rule. For example, it may be that a class must be attended by students (that is, a given instance of class must be related to at least one instance of student), but that a student might not have attended any classes (an instance of student might not be related to any specific instances of class). Such constraints could be more explicitly expressed as "a student may attend from zero to any number of classes, and a class must be attended by at least 6, but not more than 30 students." Such a statement of the relationship with the associated entities and constraints constitutes one category of business rule which is identified during the data modeling effort. This rule reflects the business practice of canceling classes for fewer than 6 students and of not registering more than 30 students to keep classes at a manageable size.

Some sample business rules are:

"The Customer Account Number on the Customer Order must belong to a Customer in the Customer file."

"A Customer Record cannot be deleted until all outstanding Invoices for that Customer are paid."

"The effective date of a promotion action cannot be earlier than the current processing date."

Note in particular that the last rule would prohibit processing retroactive promotions. This reflects one choice of business practice from among many alternatives. The key to writing effective business rules is to accurately capture the practices that are most appropriate for the function being studied.

The establishment of candidate identifiers also represents a type of business rule. Where Invoice Number is established as an identifier of the Invoice entity, it represents the business rule that no two Invoices may have the same Invoice Number.

Business rules will be captured and described as part of the data model. They may be expressed in the form of relationships, candidate identifiers, or simply as narrative statements.

Task 3.2.2.6 Develop Entity Relationship Diagram (ERD)

Purpose: To develop the entity relationship diagram.

Outputs:

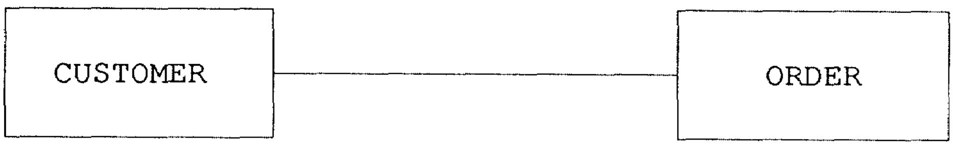
- o Entity relationship diagram

Relationships: The entity relationship diagram will become part of the composite information requirements of the function.

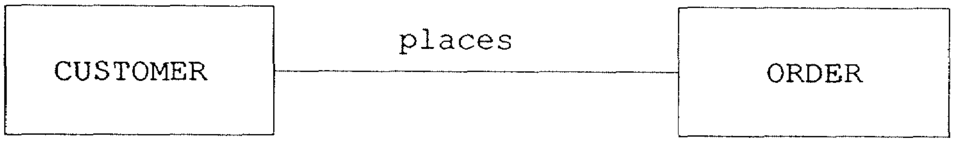
Approach: The entity relationship diagram is created by drawing a graphic representation of entities and entity subtypes (shown as boxes) connected by relationships (represented as interconnecting lines). Detailed conventions for creating entity relationship diagrams will be covered in formal training. Brief descriptions of the symbology and sample documents are shown on the following pages.

Entity Symbol - A rectangular box used to represent an entity on the diagram and containing the name of that entity.

Relationship Symbol - A straight line between two entities representing a relationship between those entities.

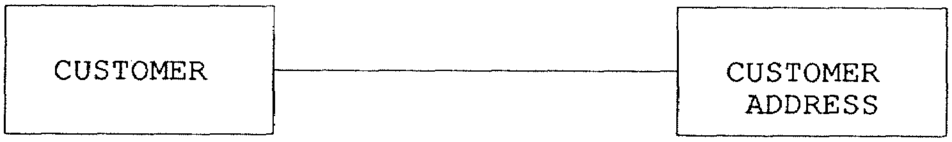


Relationship Name - A name indicating the nature of the relationship. The name will normally consist of a verb or verb phrase (in the case of a relationship between an entity and its subtypes the relationship will be expressed in the form of the criteria which is used to categorize the subtypes). The relationship between Customer and Order might be called "places" to indicate that the Customer places Orders.



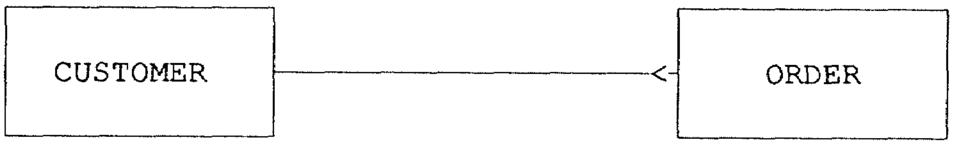
Cardinality Indicator - A measure of the number of instances of one entity that are related to a second entity. For example, not only are Customers related to orders, but a single customer may be related to multiple orders. Cardinality is shown on an ERD by the use of the "crowfoot" on either end of the relationship symbol.

Cardinality relationships can be "one to one" (Customer to Customer Address), "one to many" (Customer to Order), or "many to many" (Order to Product). "Many to many" relationships are generally resolved to two separate "one to many" relationships by the creation of a new intermediate entity.



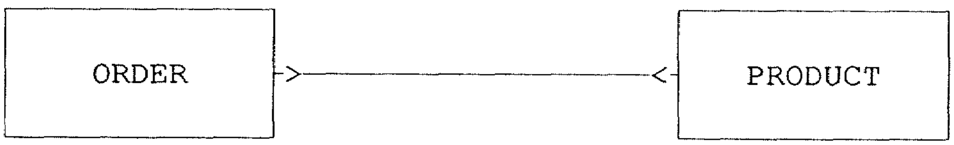
One to One

A given Customer is related to one Customer Address, and each Customer Address is related to one Customer.



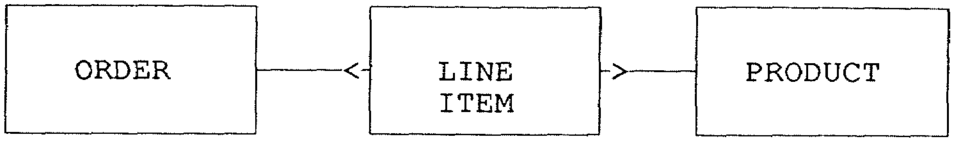
One to Many

A given Customer may be related to many Orders, but each Order is related to only one Customer.



Many to Many
(unresolved)

A given Order may be related to many Products, and a given Product may be related to many Orders.

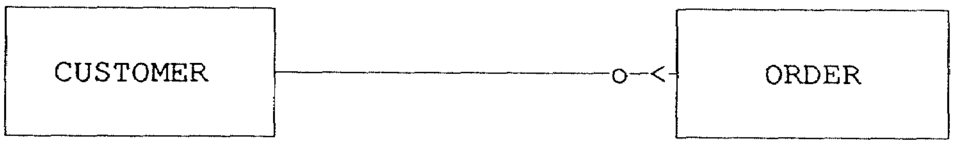


Many to Many
(resolved)

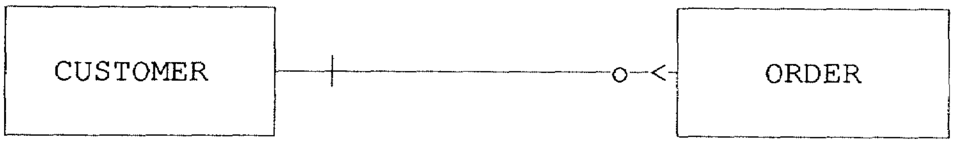
A given Order may be related to many Line Items, and a given Line Item may be related to only one Order. At the same time, a given Product may be related to many Line Items, and a given Line Item may be related to only one Product.

Cardinality can also be used to express business rules. The example above suggested that an Order could be related to many Line Items. An alternative business rule would be that an Order can be for only one Product (the business deals only in big ticket items and chooses to do business this way). In this case the cardinality should be expressed accordingly.

Optional Relationship Indicator - Indicates that a single instance of one entity may or may not be related to any instances of a second entity. An optional relationship is indicated by a small circle on the end of the relationship line closest to the entity that is optional. The following example illustrates a situation which allows for new Customers who might not be related to any Orders.

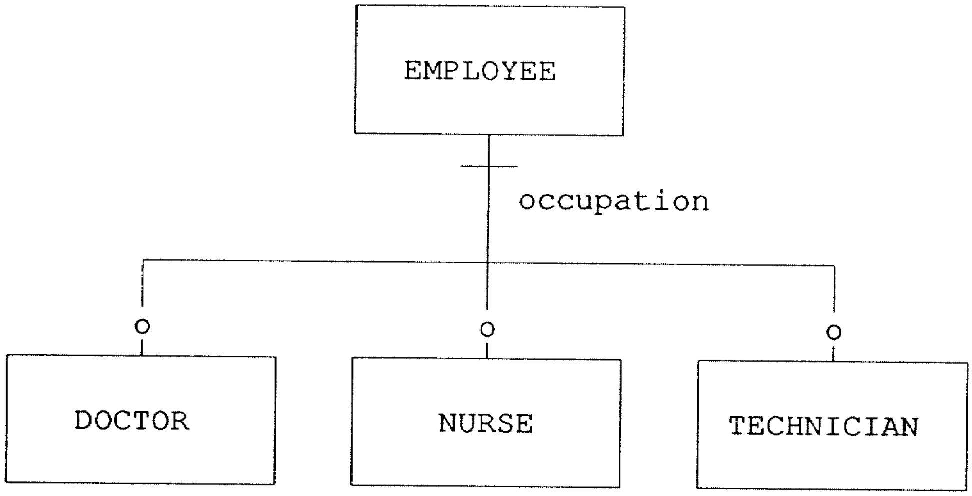


Mandatory Relationship Indicator - Indicates that an instance of one entity cannot exist without a relationship to at least one instance of a second entity. A mandatory relationship is indicated by a bar which intersects the relationship line and is perpendicular to it. The example below shows that while a Customer may not be related to any Orders, it makes no sense to have an Order that is not related to a Customer.

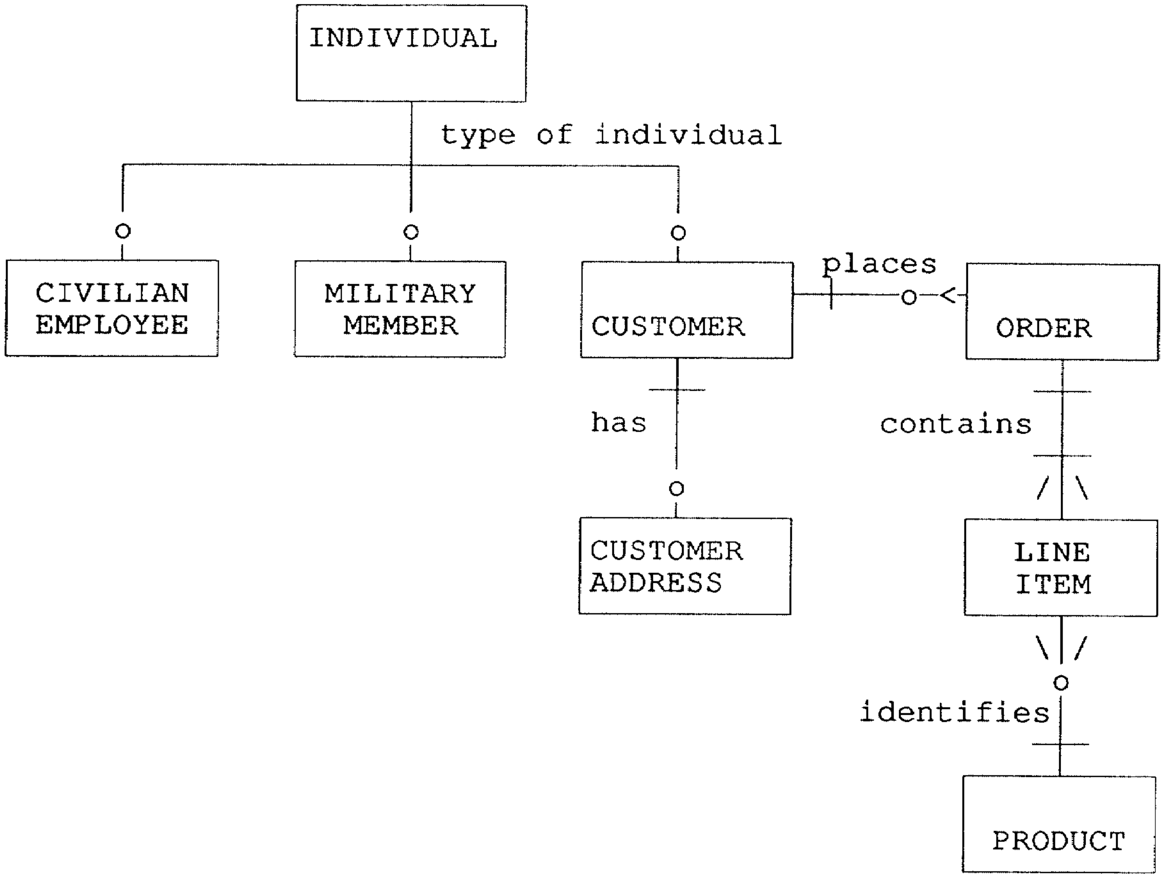


Subtype Relationship - A relationship between levels in an entity/entity subtype hierarchy is expressed as a branching line which extends from the higher level of the hierarchy and branches out to each of the lower level subtypes.

When examining subtype relationships each instance at the higher level can belong to only one of the lower level categories. It would be normal to see the mandatory indicator next to the higher level, indicating that each entity subtype must be a subtype of the higher level entity. One would also expect to see the optional indicator next to each of the lower level subtypes, indicating that the higher level may belong to any one of the subtypes.



Entity Relationship Diagram - The following ERD shows all of the symbology we have described on a single sample document.



The ERD shows an entity/entity subtype hierarchy of Individual categorized as Civilian Employee, Military Member, or Customer based on the attribute "type of individual."

The subtype Customer is, in turn, related to the entity Customer Address. A Customer may or may not "have" a Customer Address, but a Customer Address must be related to a Customer.

Customers may "place" any number of Orders (or none at all) and Orders must be placed by a Customer. Orders must "contain" some number of Line Items. Each Line Item "identifies" exactly one Product, but a Product may be identified by none or any number of Line Items.

Task 3.2.2.7 Integrate Data Across Functions and Paths

Purpose: To check cross-functional data definitions and requirements.

Outputs:

- Revised entity/entity subtype structures
- Revised attribute names and definitions
- Updates to the Corporate Data Dictionary

Approach: The approach to defining DoD's corporate information systems requirements includes subdivision by function at the highest level. A second subdivision was made by approaching definition of these functions across three parallel paths: the Future Functional Path, the Current (or Composite) Functional Path, and the Existing Information System Path.

While the processes for each function are generally unique to that function, and while they may further vary across paths, the data which these processes manage is shared across the corporation. The Individual Social Security Account Number, for example, is the same data element regardless of whether the function using it is Payroll or Contract Management. It is also the same whether we are talking about the future function, the current composite function, or an existing information system. As we progress through the data modeling effort we must ensure that shared data is described consistently throughout DoD. In this task, the Functional Group data administrator reviews the identified data requirements with the information architecture staff to ensure cross-functional data integrity.

To achieve effective data management across the organization, we must first achieve effective management of metadata to avoid redundant metadata descriptions and the need to retrofit metadata.

The definitions and descriptions of data are commonly referred to as "metadata." While the data is not owned, metadata is. Metadata is owned by an authoritative data steward. The data steward is the individual in an organization who is considered expert in the data and is capable of authoritatively defining the metadata for the particular data in question. The same data steward has responsibility for a given data attribute both across functions and across paths. Generally the data steward is involved with the business function which is responsible for the existence of the data.

It is only in this context that the steward owns the data. While the entire organization can share the data itself, everyone must use the metadata defined by the owner, or data steward.

Included in the category of metadata are data attribute definitions and entity subtype structures. The data steward for each entity should be identified as early as possible in the data modeling process.

A critical step in effectively managing metadata has already been described by the use of a standard Corporate Data Dictionary. By defining data attributes in a common dictionary, using common naming standards, and placing the data dictionary under the control of a corporate data administrator, we have assured that each function views the data consistently.

Data attribute names and definitions must be consistent both across functions and across paths. This cannot be accomplished without the use of an automated and centralized Corporate Data Dictionary.

It will not be unusual to discover that some data attributes have names which preexist the data modeling effort. Individual Social Security Account Number has been commonly referred to as SSN since long before this effort began. Such preexisting names will be particularly common in the composite function path and the existing information systems path. Where such attribute names do not meet current naming standards, a new name must be created. The old name can then be established in the dictionary as an alias for the standard name. Aliases do not have their own definitions, but can be used to reference the standard definition.

A second area where consistency must be maintained is in the creation of entity/entity subtype structures. In our earlier example we categorized Individuals as Civilian Employees, Military Members, and Customers. Individual could just as easily be categorized as "living or dead", "male or female", or by state of birth. Which method of categorization is most appropriate depends upon the perspective of the function being described. Different functions may have different perspectives, and therefore, would tend to create different subtype structures. However, it is the "owner" of the data who must make the determination as to which structure is most appropriate for corporate-wide use.

For example, if we determine that the personnel function owns Individual, it is personnel which must define the entity subtype structure. If the financial business function needs to use the data, that function must use the structure defined by personnel and cannot create its own. If the financial group feels that their information requirements cannot be satisfied by the structure provided, a solution must be negotiated with personnel. Financial will then use the subtype structure defined by Personnel in their Financial data model.

Identification of shared entities is accomplished during integration activities. Entities are generally shared in the sense that an entity is managed (created) by one function and referenced by another. Shared entities will be part of the external interface of the referencing function. Where shared entities are discovered, the group must investigate the function which owns the entity and adopt subtype structures as appropriate. Owning functions may be identified by tracing entities via the external interfaces through which the data is acquired by the function.

Entity/entity subtype structures must be consistent among functions within a path. However, unlike data elements, these structures do not have to be consistent across paths (that is, between the future model and the composite models for the same function). Since these structures represent a way of looking at the organization of data from a functional perspective, and since each path may describe a different functionality (the future function may not be the same as the current function), it is acceptable that these structures be different across paths.

Most overlaps among functional groups will be identified during integration activities and will be resolved through negotiation.

Duplicating entities and entity subtype structures and their descriptions in multiple data models as we have described is recognized as being redundant. However, to fully represent the information requirements of individual functions, such redundancy is necessary. Redundancy of the effort involved in creating these definitions is <u>not</u> necessary. Where one function uses data owned by another, that function should reference the metadata documented by the owning function rather than expend resources in creating its own.

The Functional Group data administrator will help the group identify any cross-functionsl inconsistencies in data attribute names, definitions, or any other elements of the data model. If inconsistencies exist, the group will revise the entity/entity subtype structure and attribute names and definitions. Any updates to the Corporate Data Dictionary will be made at this time.

STEP 3.2.3 COMPOSITE HIGH-LEVEL MODEL RECONCILIATION

Purpose: To reconcile the action diagrams and supporting information views with the high-level data model for the purpose of ensuring the completeness of the two models and of maintaining consistency between them.

Description: In this step the entities and entity subtypes identified in the data model from Step 3.2.2 will be reconciled with the action diagrams from Step 3.2.1.

A subprocess to entity CRUD matrix will be developed. Aggregated action diagram information views will be compared to the matrix to ensure that the data model satisfies each information processing requirement at the subprocess level.

Outputs:

- Subprocess level information views
- Subprocess to entity (CRUD) matrix
- Revised data model including
  - New entities and entity subtypes
  - New entity and entity subtype descriptions
  - New entity and entity subtype relationships
  - Revised entity relationship diagram
  - Newly identified attributes with descriptions
- Revised process model
- Updates to the Corporate Data Dictionary

Relationships: The data model that is output from this step will facilitate an evaluation of the ability of existing systems to meet future and composite information needs and can be used as a basis for strategy decisions regarding transition from composite to future functions.

The revised data model will be fully attributed in Step 3.2.5. The revised process module will be described to the module level in Step 3.2.4.

Task 3.2.3.1 Aggregate Action Diagrams to Subprocess Level Information Views

Purpose: To aggregate the action diagram level information views into subprocess level information views.

In Step 3.2.1 subprocesses from Phase II were partitioned into macro-level action diagrams and an information view was developed for each action diagram. The overall purpose of Step 3.2.3 is to ensure that the information requirements of the action diagrams can be satisfied by the data model. The specific purpose of this first task in that step is to consolidate the number of information requirements into a manageable number.

Outputs:

- o Subprocess level information views

Relationships: Aggregated subprocess level information views will be used to build the CRUD matrix in Task 3.2.3.2.

Approach: Given the significant number of action diagrams which may be needed to represent a large system, creating an entry on a CRUD matrix at the level of action diagram would result in a document of unmanageable size. For this reason the information views will be aggregated back to the level of subprocess (as identified in Phase II). The process model entries for Task 3.2.3.2 will then be at the level of subprocess. The corresponding view will represent the sum of the information views for all of the action diagrams within a given subprocess.

For example, assume that we identify three action diagrams for a subprocess which we will call Subprocess One. In turn we would identify an information view for each action diagram. The view for Action Diagram 1 might be a reference to Entity 1. The view for Action Diagram 2 could consist of a reference to Entity 2 and an update of Entity 3. Finally, the view for Action Diagram 3 might be to create, reference and delete Entity 3. These three views could be aggregated to form a single view consisting of references to Entities 1 and 2 and a create, reference, update, and delete of Entity 3. A data model which satisfies the aggregate view would, by definition, satisfy the individual Action Diagram views.

It may appear that the effort to identify action diagram level views, only to aggregate them later, was redundant. In fact this detailed understanding of information requirements is necessary to fully understand the data requirements of the subprocesses. Further, the detailed action diagram level views will be used later in Step 3.2.4.

Task 3.2.3.2 Develop Subprocess to Entity (CRUD) Matrix

Purpose: To facilitate the validation of the data model against the process model by showing on a single document all entities and entity subtypes arrayed with all subprocesses.

Outputs:

- o Subprocess to entity (CRUD) matrix

Relationships: The CRUD matrix will be used to validate information views in Task 3.2.3.3 and to validate the management of entities in Task 3.2.3.4.

Approach: A matrix is created showing subprocesses from Task 3.2.3.1 (aggregated action diagrams from Step 3.2.1) on the vertical axis and data entities from Step 3.2.2 on the horizontal axis.

The nature of the relationships between subprocesses and entities are then expressed in the context of the action(s) that the subprocess may take on each entity. Potential actions are:

C Create: subprocess creates instances of the entity

R Read or Reference: subprocess references instances of the entity

U Update: subprocess modifies attributes of an instance of the entity

D Delete: subprocess may remove instances of the entity

For each intersection of a subprocess and an entity on the matrix the group must consider potential action(s). If there is no action the intersection remains blank. Otherwise one or more characters are entered at the intersection as appropriate.

Since this Matrix is critical to the completion of subsequent tasks in this step, we will briefly discuss interpretation of the following sample matrix.

Sample Subprocess to Entity (CRUD) Matrix

(Ent = Entity, C = Create, R = Reference, U = Update, D = Delete)

	Ent 1	Ent 2	Ent 3	Ent 4	Ent 5	Ent 6	Ent 7	Ent 8
Subprocess 1	C							
Subprocess 2		R		R		R	R	
Subprocess 3			CRUD					
Subprocess 4							CUD	
Subprocess 5		R		CRUD				
Subprocess 6								
Subprocess 7		R				U	R	
Subprocess 8	R	CRUD						R
Subprocess 9							R	

The horizontal line extending to the right from each subprocess is one representation of the information view of that subprocess. In the sample CRUD matrix above the view of Subprocess 2 consists of references to Entities 2, 4, 6, and 7. The information view of Subprocess 7 includes reference to Entities 2 and 7 and the potential updating of Entity 6.

The vertical line extending beneath each entity represents management of that entity within the function. Entity 2 is created, read, updated and deleted by Subprocess 8 and merely referenced by Subprocesses 2, 5, and 7.

Columns that contain only "R"s indicate that the function being documented only references the data and does not manage it beyond that. In such cases the entity is being managed by some other function, i.e., the data is part of the external interface of the function being modeled.

The information view for Subprocess 6, which does not act on any entities, is an obvious problem. Virtually every subprocess will have some type of information view. Either the data requirements of the subprocess are not properly understood, the information view is not needed, or another entity needs to be added to the matrix. Entities will be added only after careful verification that the data model is incomplete. First consideration will be given to potential scoping problems. It may be that the process in question exists in some current function, but may not belong in the composite function. In this case the process itself will be removed and potentially transferred to the another business function.

Similarly, the column under Entity 5, which shows that no subprocesses act on Entity 5, needs to be investigated. If the data entity is not referenced by any subprocesses within the function, then either there is no need for that entity to be included in the data model for this function, or a subprocess which does manage the entity has been overlooked and must be added to the process model. Again, careful consideration must be given before removing entities from the data model. Perhaps the scope of the composite function, as contrasted with the scope of some current function, was not properly considered in the process model.

Removal of an entity from the data model of one particular function does not imply that it does not continue to exist on data models of other functions.

Discrepancies such as those described above are a natural part of the reconciliation process. They are simply one step in the reconciliation of the process and data modeling efforts which have proceeded along parallel, but independent, paths.

Since this subprocess to entity matrix will be of considerable size and may require considerable manual effort, automated support is considered essential.

While our matrix tells us which entities are needed to satisfy the information requirements of each subprocess, it does not resolve any questions we may have at the level of data attributes. To be fully understood, information views must be expressed at the attribute, or data element, level. However, we have not yet fully defined our data to the attribute level nor have the information needs of our subprocesses been consistently identified in such detail. This level of validation must wait until Step 3.2.6.

Task 3.2.3.3 Validate Information Views Against the Data Model

Purpose: To ensure that each of the information views identified in process model action diagrams can be satisfied by the data model.

Outputs:

- Revised data model including
  - New entities and entity subtypes
  - New entity and entity subtype descriptions
  - New entity and entity subtype relationships
  - Revised entity relationship diagram
  - Newly identified attributes and descriptions
- Revised process model
- Updates to the Corporate Data Dictionary

Relationships: Validated information views will become part of the <u>composite</u> process model and will be used in Step 3.2.5 to develop detailed information views (entity attributes). The validated data model will be fully described in Step 3.2.5.

Approach: In Step 3.2.1 the group developed an information view for each action diagram. These information views were defined from a process perspective and reflect the data needed by the associated action diagram to execute its task. While in some cases individual data elements may be included in the information views, the data model itself is not yet fully attributed and views may be reliably validated only to the entity level.

Validation of views may be accomplished by examining the horizontal line extending from each subprocess. If the CRUD indicators match those defined in the information view, no action is necessary. Where they do not match, it is an indication that corrective action may be necessary.

The group will first verify that the information view and the data model terminology is expressed consistently and that the problem is not simply one of semantics. Strict application of the corporate naming standards should eliminate the potential for this problem.

Where the information view is appropriate and correct, and is not satisfied by the data model, it is an indication that the data model itself is incomplete. In this case the data modelers will take appropriate action to add any needed entities and entity subtypes, complete with relationships and descriptions, to the data model.

Task 3.2.3.4 Verify Entity Management

Purpose: To ensure that all entities are created, referenced, updated and deleted as appropriate within the function.

Outputs:

- o Revised data model
- o Revised process model
- o Revised subprocess to entity (CRUD) matrix

Relationships: The data model from Step 3.2.2 and the process model from Step 3.2.1 will be validated in this step. These models will be fully described in Steps 3.2.4 and 3.2.5.

Approach: The CRUD matrix will be examined to ensure that all entities are "managed" appropriately. That is, the group will verify that processes exist which appropriately create, reference, update and delete each of the entities in the data model. Entity management problems must be resolved. If the Materiel Management function needs to reference the Item entity, and no process can be identified which creates Items, then there will be no Items to reference. Entity management may occur either in the function being studied, or in some other function. Verification is accomplished by examining the vertical columns extending beneath each entity.

If there are no indicators in a column it is evidence that the entity may not be needed by the function or that a process within the function has been overlooked. If no process can be discovered that at least references the entity, the entity should be removed from this data model and perhaps added to the data model for another function.

If there are no create indicators in a column it indicates that the entity is imported from an external function. This will be verified by coordination with the external source of the data to ensure that the required data will be available to support the function when needed. The group must ensure that a process to create the entity already exists or will be established either in this or some other business function.

STEP 3.2.4 COMPOSITE DETAILED PROCESS MODEL

Purpose: To refine the macro-level action diagram into a detailed action diagram and clearly represent processing control structures, data actions, and data elements. The output will provide the system designers with a detailed view of the processing requirements of the business function.

Description: This step develops a detailed action diagram by reviewing the macro-level action diagram and adding process control tools from the expanded action diagramming repertoire. Through the inclusion of these additional action diagramming symbols, emphasis is placed upon identifying major processing modules, opportunities for concurrence, data actions, and basic data elements. As these processing control structures are introduced, a series of business practices will be documented which detail "how" to perform processing.

- A business practice is an expression of "how" to manage and execute an aspect of a functional activity.

Outputs:

- Detailed action diagram showing data actions
- List of data elements
- List of business practices

Relationships: The basis for this step is the macro-level action diagram produced by Task 3.2.1.1. The action diagram produced in this step will be further refined in Step 3.2.6. The list of data elements will be integrated with the list of data attributes generated by the data modeling group. This process to attribute matrix will assure data integration across the process and data modeling analyses. After the data integration subtask a consolidation review with the other functional groups is recommended.

Task 3.2.4.1 Composite Process Flow Analysis

Purpose: To further refine the macro-level action diagram to reflect sequence control and procedures.

Outputs:

- Action diagram showing detailed process logic

Relationships: The basis for this step is the macro-level action diagram produced by Task 3.2.1.1. The action diagram produced in this step will be further refined in Step 3.2.6.

Approach: This task further refines the macro-level action diagram into a detailed action diagram. This is done by reviewing the macro-level action diagram and adding process control symbols from the expanded action diagramming repertoire. Emphasis is placed upon identifying major processing procedures (called modules), opportunities for concurrence, and shared processing modules.

The action diagram will be enhanced by adding the concept of modules and concurrence flow control. Figure 1, below, graphically illustrates the representation of each of these diagramming constructs. The macro-level action diagram produced by Task 3.2.1.1 is reviewed in two passes to identify processing logic details. The first pass emphasizes the identification and categorization of the action modules. The second pass looks for opportunities for concurrent processing and direct changes in processing flow.

Three different types of modules should be identified in this step.

- A unique processing module is an activity component of an action which does not occur elsewhere in the function or only occurs elsewhere with significant processing changes (as in differing service business practices). For example, a module to "produce pay check."

- An undefined processing module is a collection of unknown (by the work group) processing activities which will be defined by the design group at a later date. For example, a module to "electronically send contract data."

- A common processing module is a recurring set of processing activities which are used in several locations in performing the processing function. For example, a module to "perform error handling."

The concurrence tool affects the processing flow. It can be used to simplify the logic and may allow the designer some additional flexibility when trying to streamline the processing.

- o Actions and modules which should be considered for concurrent processing are those which do not have any inherent sequential order of precedence relative to the surrounding processing actions, i.e., actions or blocks of actions which can be accomplished at the same time as some other action or block of actions.

Subtask: Identify Processing Modules

The macro-level action diagram must be reviewed to identify unique, undefined, and common processing modules. Previously the actions comprising the process or subprocess were identified. Often these actions either represent a set of processing activities and are already modules or can be logically grouped and replaced with a module. If grouping common sets of actions simplifies the logic, use of modules (which should be defined as specifically as possible in the documentation) is justified.

As modules are identified or created, a description containing a mini-action diagram and narrative explanation should be prepared. Once the macro-level action diagram has been defined at the module level it is referred to simply as an action diagram. This terminology will be used during the next final stages of process decomposition.

Subtask: Identify the Sequence Control Structure

Having identified the processing steps to the module level in the action diagram, it is now possible to analyze opportunities for concurrent processing and restructuring of the process flow.

Not all opportunities for concurrence need to be taken or identified. Attempts to increase concurrence may disrupt the underlying logical flow of the process, causing needless complexity. An opportunity for concurrence is a concurrent process which does not disrupt the logical flow of the function process. For example, in the Civilian Pay function, the process "calculate deductions" may require sub-processes of (1) identify health benefits deductions, (2) identify Combined Federal Campaign donations, and (3) identify life insurance allotments. If these can be accomplished in parallel, they could be identified as concurrence opportunities. Concurrence should be viewed as a way of streamlining the overall function processing and thereby simplifying its structure.

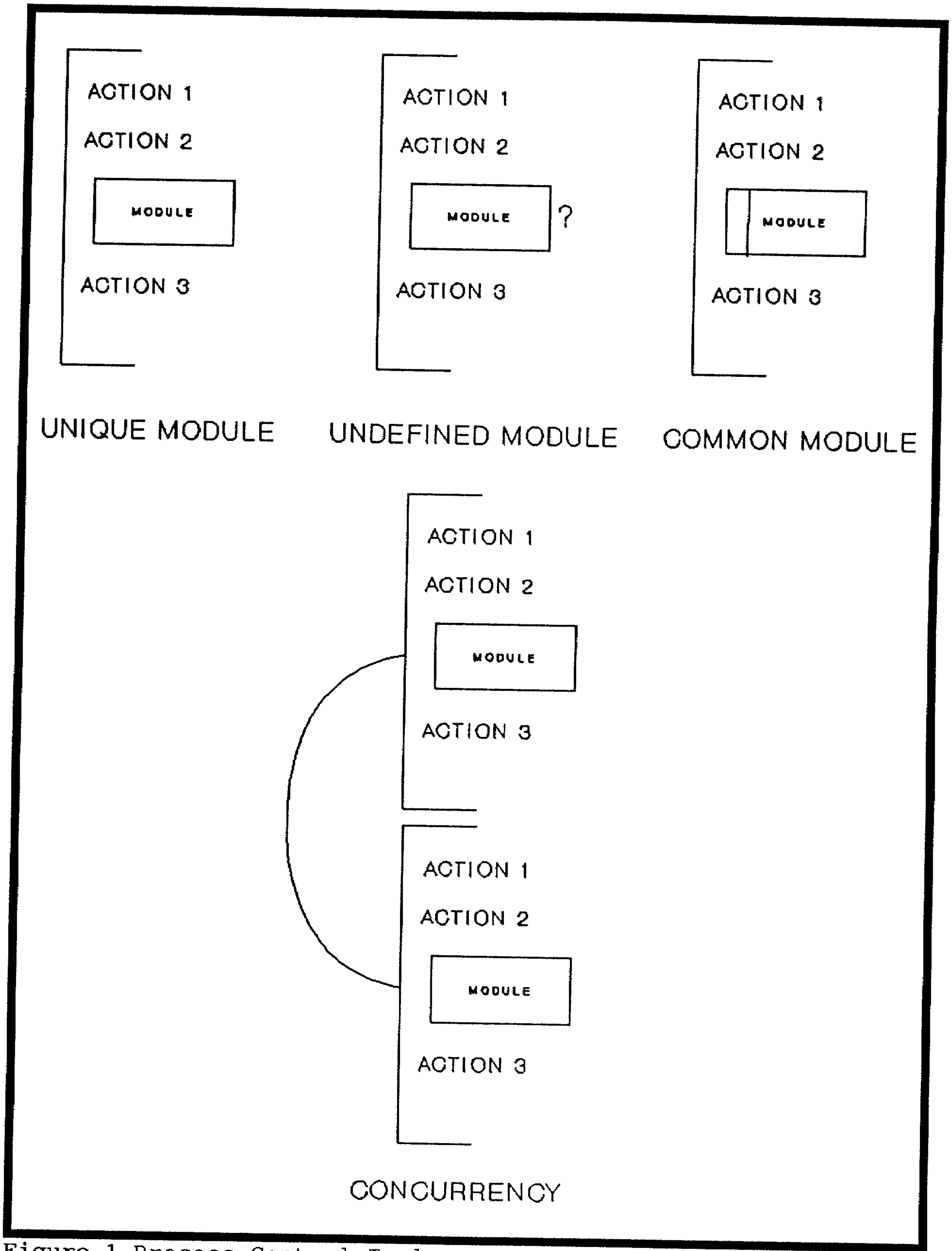


Figure 1 Process Control Tools

Task 3.2.4.2 Detailed Composite Information Processing Analysis

Purpose: To identify the detailed data actions performed at the module level. It will identify processes which use data, and where appropriate the method of use.

Outputs:

- o Detailed action diagram showing data actions
- o List of data elements
- o List of business practices

Relationships: The action diagram with all modules identified was produced by Task 3.2.4.1 and serves as the primary input to this task. The list of data elements will be used in Step 3.2.6.

Approach: This task adds the simple and compound data actions to the action diagram to further depict the input and output activities within the function. This is the most detailed level of action diagram produced in the analysis of the function.

The action diagram produced by Task 3.2.4.1 is reviewed in two subtasks to identify input and output processing details. The first pass emphasizes the identification of simple data actions and the second subtask emphasizes compound data actions.

- o Simple data actions are elementary data activities which include Create, Read, Update, and Delete (CRUD) processing. In the action diagram they are represented by a line containing one of the CRUD key words followed by a single line box identifying the information view being acted upon.

- o Compound data actions are complex data activities which for this analysis will only include SORT and JOIN. In the action diagram they are represented by a compound data action key word, followed by a double line box containing the information view being acted upon, followed by a condition phrase such as BY, IF, or WHEN.

The data actions are extensions to the action diagram tools repertoire and completes the set of diagramming tools. Figure 2, below, graphically depicts these tools. Completion of this level of action diagramming represents the final step of the process decomposition methodology.

After bringing the action diagram to the final level of detail, the data element list produced by Task 3.2.2.3 will be reviewed and updated. This updated list represents the maximum level of data element detail attainable by the process work group and will be integrated with the analysis performed by the data modeling group in the next step.

Subtask: Identify Data Actions

The data diagram should be reviewed for actions which are or require data activities. Typically, the simple CRUD data actions will suffice. However, some processes will require compound data actions. These occurrences should be carefully considered to eliminate design-like influences from the inherent, processing required instances. It is critical to only identify the "what" aspects of the data activities and to resist specifying any "how" design approaches. Most often this analysis will involve the conversion of already identified actions and modules to a more specific data activity structure, such as changing READ CONTRACT-STUFF to SORT CONTRACT-STUFF BY DATE-RECEIVED.

It is not unusual for some confusion to arise over the SORT and JOIN verbs as used in this type of analysis, when compared with their use in design. For this analysis effort <u>the SORT action identifies occasions when the data inherently must be in a specific order.</u> It is not an attempt to influence the flow of processing, but rather a statement of a requirement. In the contract example above the contracts might require processing in date received order because of a business practice - <u>not</u> because it is more efficient or easier.

<u>The JOIN verb identifies processing actions which require data external to the function being analyzed.</u> For example, if the Contracts Payment analysis has a process requiring data maintained by the Contracts Management area, a JOIN could be used to signify this requirement. It could look like JOIN AUTHORIZED-PAYMENT WITH VENDOR-ADDRESSES BY VENDOR-IDENTIFIER. In this example, AUTHORIZED-PAYMENT is known to exist within the Contracts Payment area; however, the VENDOR-ADDRESSES are part of the Contract Management area. The JOIN signifies that this process requires data controlled by another function.

The use of SORT and JOIN tools is intended to add to the information being collected and depicted in the analysis. As their application is not always required or not always obvious, care must be taken to assure that they are not being used as design tools. SORT and JOIN can provide details on business practice requirements and integration requirements when used in the analysis. They typically will act upon information views but also can identify data elements performing special roles like keys (data controlling record order) or indices (data for linking or looking up related data). This compound data action will be useful to the designers responsible for creating the final system.

Subtask: Generate a List of Process Required Data Elements

After completing the data action decomposition, the action diagram should be reviewed for modifications and additions to the data element list created by Task 3.2.1.2. The addition of the data action structures to the diagram should serve to clarify some of the information requirements and data dependencies in the process. These insights may cause new or modified information views to be created and new data elements to be identified. These elements should be added to the Task 3.2.1.2 list for use in the Step 3.2.6 reconciliation analysis.

Subtask: Generate a List of Business Practices

As the action diagram gains more and more detail, the methods and techniques for accomplishing processing requirements will become discussion issues and will require documentation. These are the business practices of the function. They influence the sequence and control of the processing and define the normal and exception logic control conditions. As each business practice is identified, it should be documented through a brief descriptive narrative. A name or short phrase identifying the business practice should be created for ease of reference, and a comprehensive list generated.

Business practices were introduced in Task 2.1.6.2 and preliminary definitions provided. This documentation should be reviewed in light of the additional processing details now available to the work group. Business practices from Task 2.1.6.2 should be further clarified, and any additional business practices should be added to the list. For consistency the format identified in Task 2.1.6.2 should be continued.

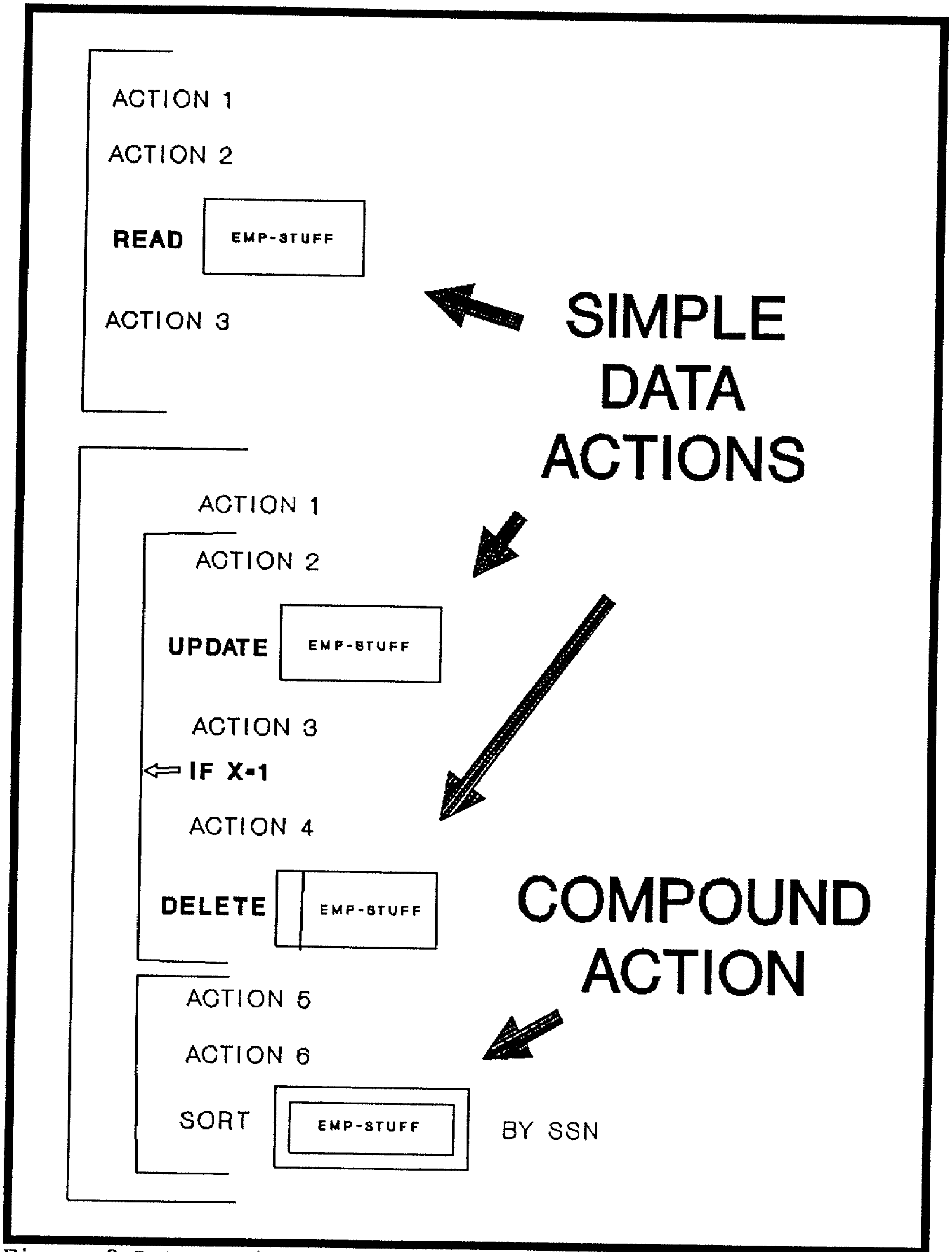


Figure 2 Data Action Tools

STEP 3.2.5 COMPOSITE DETAILED DATA MODEL

Purpose: To describe at a detailed level the data portion of the composite information requirements for the function.

Description: In Step 3.2.2 a high-level composite data model was created. This model was reconciled in 3.2.3 with the logical process model (entity diagram) from Step 3.2.1. In this step the data model will be fully attributed and carried to its final level of detail. The data model will then be normalized.

In the Phase III data modeling process the data will not be organized into independent application oriented subsets. However, the data will be organized such that it lends itself to partitioning into such subsets for the purpose of satisfying the information requirements of a variety of implementation strategies. This partitioning is not a functional group activity. It will be performed during transition to composite standard information systems.

Outputs:

- o Attribute names and definitions
- o Updates to the Corporate Data Dictionary
- o Normalized data model
- o New entities and entity subtypes, with descriptions

Relationships: The detailed data model will be used by central design activities to design physical data structures in support of applications.

Task 3.2.5.1 Identify Attributes

Purpose: To identify all additional attributes needed to fully describe the entities represented on the data model.

Outputs:

- o Attribute names and definitions
- o Updates to the Corporate Data Dictionary

Relationships: The attribute definitions will be entered in the data dictionary for reference by functional groups, central design activities, and other interested parties throughout the life cycle of the business function.

Approach: In Step 3.2.2 the group identified a limited number of attributes, including candidate identifiers. Identification of these attributes was as a fall out of the modeling process, not as part of a concerted effort to fully attribute the date model. In this step the group will make such an effort and will identify all additional attributes needed to fully describe the entities represented on the data model. In Step 3.2.2 the group was also encouraged to minimize the amount of information captured about each attribute. In this step the group will collect all relevant information. The end product of this subtask will be a set of fully attributed entities.

As in the previous step we will follow the principle of inheritance which allows us to establish attributes at the highest level of a hierarchy of entities and entity subtypes. These attributes are then inherited by each of the subtypes.

All attributes will be named in accordance with corporate naming standards and entered into the Corporate Data Dictionary.

As development proceeds, it will be common to find that metadata (names and definitions) for attributes identified in the function being studied has already been created by other functional groups. In many cases this metadata may not agree with what the group would like to use. However, it is critical to the integrity of the corporate data of DoD that the group not create its own names and definitions. Sound data management principles call for the authoritative data steward to provide metadata for each attribute. Where metadata is in dispute, the appropriate data steward will be identified and consulted.

While the data steward "owns" the metadata, the data administrator is the final authority in controlling entries into the Corporate Data Dictionary. The data administrator ensures that appropriate naming conventions have been followed and that the integrity of the data in the dictionary has not been compromised. In cases where the group may function as the data steward for selected pieces of data, they must still bow to the authority of the data administrator on data standardization and integrity issues.

The group must not create local names and definitions for specific use within a function for data which is shared across functions. Rather, the group must use legitimate corporate metadata created by others and will create metadata only for those attributes for which they are the data steward.

Attribute information will include such things as the attribute name, its description, purpose, domain values, default values, optionality, security, length, type, and ownership.

Task 3.2.5.2 Normalize the Data Model

Purpose: To further standardize the structure of the data by applying the formal rules of normalization. The process of normalization will ensure that the data model is consistent, non-redundant, stable, and free of process bias.

Outputs:

- o Normalized data model
- o New entities and entity subtypes, with descriptions
- o New attributes and definitions
- o Updates to the Corporate Data Dictionary

Relationships: Normalized data structures will be referenced by central design activities in the design of physical data structures to satisfy application level information requirements.

Approach: Throughout the data modeling process the group has been encouraged to look at the data independently of the processes, to capture the natural relationships of the data, and to assign attributes nonredundantly to their appropriate entities. As a result, the data model at this point will reflect an organization of data that is largely consistent with a normalized structure.

However, it is inescapable that the groups will view the data with some degree of process bias. Group members are selected because of an expertise in the function. This expertise is gathered through years of experience with the processes that have historically been executed to carry out the function. This experience will color the way the data is viewed and will influence how the group defines "natural" relationships.

The formal rules applied during the normalization process force adjustments to the data model which will remove most traces of process bias. These adjustments are critical to efficient operations in a modernized environment.

Although normalization is often commonly (and erroneously) regarded as a technical activity, it in fact depends on a thorough knowledge of the data and its characteristics from a functional perspective. The active participation of the group is critical.

The data model will be normalized at least to third normal form (and potentially to fifth normal form). Normalization to this level will ensure a correct, consistent, and nonredundant data model. Specifics of the normalization process will be addressed in formal training sessions.

During the normalization process additional entities, entity subtypes, relationships, and attributes may be identified. These will be documented as described in previous steps.

STEP 3.2.6 COMPOSITE DETAILED MODEL RECONCILIATION

Purpose: To reconcile the action diagrams and supporting information views (data elements) with the detailed data model at the level of data attributes. To assure consistency between process and data information requirements and ensure completeness of the data model at the attribute level.

Description: In this step the attributes identified in the data model from Step 3.1.5 will be reconciled with the subprocess level information views from Step 3.1.4 to ensure that these views can be satisfied to the element level. A process to attribute (CRUD) matrix will be developed using subprocesses (module-level data element requirements aggregated to the subprocess level) and the attributes from the entity descriptions. For purposes of this document data element and attribute are considered equivalent.

Outputs:

- o Subprocess to attribute (CRUD) matrix
- o Revised Action Diagram
- o Revised data model reflecting new attributes
- o Updates to the Corporate Data Dictionary

Relationships: This step is the end point for the detailed information system requirements definition activity. The information system requirements from this step will be prioritized in Step 3.1.8 and will become the basis for the information systems strategy. The outputs from this step will be among the documents provided to a central design activity tasked to develop future automated information systems.

Task 3.2.6.1 Aggregate Action Level to Subprocess Level Information Views

Purpose: To aggregate the detailed action level information views into subprocess level information views.

In Step 3.1.4 the action diagrams from Step 3.1.1 were partitioned into modules and data actions. An information view was developed for each process rectangle and defined in terms of data elements. The overall purpose of Step 3.1.6 is to ensure that the information requirements of the modules and data actions can be satisfied by the data model at the level of data elements. The specific purpose of this task is to consolidate the action level information requirements into a more manageable number by aggregating them to the subprocess level.

Outputs:

- o Subprocess level information views expressed in terms of data elements

Relationships: Aggregated subprocess level information views will be used to build the CRUD matrix in Task 3.1.6.2.

Approach: Action level data element requirements are aggregated to the subprocess level to eliminate redundancy and consolidate requirements. Given the large number of actions which will be required to represent a large system, creating an entry on a CRUD matrix at the level of the action would result in a document of unmanageable size. For this reason the information views will be aggregated back to the level of subprocess. A similar task was performed in Step 3.1.3 to aggregate action level information views to subprocesses. This task will proceed along the same lines and will result in aggregated views for the corresponding subprocesses. This is a refinement of the existing information views to reflect the more detailed data requirements portrayed in the action diagram. The resulting consolidated views will represent the sum of the data elements required to satisfy all of the actions within a given subprocess.

A simple way of collecting the data element information is by creating two working lists at the subprocess level of detail, one reflecting input data requirements and the other showing output data produced. These data elements are shown in the action diagram at the top right of the process rectangles for input, and the bottom right of the process rectangles for output. The lists should not duplicate the names of the data elements, even though they are used more than once by the actions or modules within a subprocess.

For example, assume that we identify twelve modules as part of three different process rectangles within a given subprocess. Each of these modules would have an information view expressed in terms of data elements. Any data element which was part of an information view for any of the modules would become part of the information view for the subprocess. The resulting subprocess information view will be defined in terms of data elements.

The identification of information views at the action level is essential to understanding the data requirements of the processes at the detailed level. The consolidation of these information views into a single subprocess level view is equally important in order to be able to deal with the inherent complexity at a reasonable level.

Task 3.2.6.2 Develop Subprocess to Attribute (CRUD) Matrix

Purpose: To develop a matrix that arrays all subprocesses against all of the attributes from the data model. This matrix is a refinement of the subprocess to entity matrix created in Task 3.1.3. It will allow for the reconciliation of the action-level data requirements (aggregated to the subprocess level) with the data attributes.

Outputs:

- o Subprocess to attribute (CRUD) matrix

Relationships: The subprocess to attribute matrix is a reconciliation tool which is the final level of reconciliation of the process model to data model. The matrix will be used to validate the availability of data elements in Step 3.1.6.3 and to validate the management of data elements in Step 3.1.6.4.

Approach: A matrix is created showing subprocesses on the vertical axis and data attributes on the horizontal axis. The nature of the relationships between subprocesses and attributes are then expressed in the context of the action(s) that the subprocess may take on each attribute. Potential actions are:

C Create: subprocess creates the data

R Read or Reference: subprocess references the data

U Update: subprocess modifies the data

D Delete: subprocess removes the data

For each intersection of a subprocess and an attribute on the matrix the group must consider potential action(s). If there is no action the intersection remains blank; otherwise, one or more characters are entered at the intersection as appropriate. The horizontal line extending to the right from each subprocess is one representation of the information view of that subprocess. The vertical line extending beneath each attribute represents management of that attribute within the function. Columns that contain only "R"s indicate that the attribute is probably owned by some other function and is part of the external interface of the function area being modeled. Since this document will be of considerable size and may require considerable manual effort, automated support is strongly recommended.

A sample CRUD matrix was included with Step 3.1.3. While the matrix from Step 3.1.3 arrayed subprocesses with entities, rather than with attributes, interpretation is similar.

Task 3.2.6.3 Verify Information Views With the Data Model

Purpose: To ensure that the consolidated action level information views identified can be satisfied by the data model.

Outputs:

- Revised data model
- Revised process model
- Updates to the Corporate Data Dictionary

Relationships: This task makes use of the information views developed by the process decomposition analysis (as revised in task 3.1.6.1) and the detailed CRUD matrix. Reconciled information views will become part of the final future process model.

Approach: In Step 3.1.4 the group developed an information view for each module or data action acting upon data. These information views were defined from a process perspective. They reflect the data needed for the associated action to execute its task. The subprocess level information views represent the data needed by all of the included actions within the subprocess to perform their tasks.

Reconciliation of views may be accomplished by examining the horizontal line extending from each process. If the CRUD indicators are consistent with the definition of the information view, then no action is necessary. Where they do not, corrective action may be necessary.

It is particularly at the attribute level that the problem may be one of semantics. This occurs where the process model refers to a data element by one name and the data model refers to the corresponding attribute by another name. In such cases the inconsistency should be isolated and the appropriate name used in both locations. Problems of this nature will be minimized where an effective data administration program is in force and where the Corporate Data Dictionary is used for both models.

It is also possible that data elements required by a subprocess may be "derived" data elements.

- Derived data elements are data elements whose values can be determined based on the values of other related data elements.

Where the data elements needed to derive the desired value are present in the data model, a problem does not exist. The appropriate definitions will be created in the data dictionary and the relationship among the derived element and the elements used to derive its values documented.

Also, the process model should be reviewed to determine the feasibility of modifying the process model to use different data elements to achieve the desired result. If this is readily done, the process model should be revised to reflect the new method.

Where inconsistencies are not resolved as described above, either the data model or the process model may need to be modified.

If it is determined that a required information view references an attribute that does not exist, the group will investigate to ensure that the subprocess, or any particular component actions and modules, belong within the future scope of the business function. Subprocesses, actions, or modules may be eliminated or moved to other functions as a result of this investigation. This may be a cross-function integration issue which must be coordinated with the affected work group.

If it is determined that the attribute is in fact required by the function, the data modelers will take appropriate action to add the attribute, complete with relationships and descriptions, to the data model.

Attributes which are not referenced by any subprocess should be investigated for potential missing processes or possible removal from the data model.

Since these same subprocesses were reconciled with entities in Step 3.1.3 it would be unusual to discover that a subprocess did not act on any attributes. If this should occur, the group will revisit the Step 3.1.3 Subprocess to Entity matrix and determine which entities the subprocess acted on. The group will then identify which attributes of that entity are of interest to the subprocess and ensure that these are included in the data model.

Task 3.2.6.4 Verify Attribute Management

Purpose: To ensure that all attributes are created, referenced, updated and deleted as appropriate within the function.

Outputs:

- o Revised CRUD matrix
- o Revised data model
- o Revised process model

Relationships: The revised data and process models from this step will be packaged in Step 3.1.7 and will become the major part of the information systems requirements for the function.

Approach: The CRUD matrix will be examined to ensure that all attributes are managed appropriately. This is accomplished by examining the vertical columns extending beneath each attribute.

If there are no indicators in the column it is evidence that the attribute may not be needed by the function. This should be discussed by both the data and process modelers. The future scope should be considered in deciding whether management or reference of an attribute is appropriate for the business function. If a process has been overlooked, it should be fully documented and added to the process model and the CRUD matrix. If it is discovered that none of the subprocess (or their component actions and modules) reference the attribute, then the attribute should be removed from the data model.

The lack of Create indicators for an attribute indicates that the attribute may be imported from an external function. This should be verified through reference to the detailed action diagram and by contacting the external source of the data to ensure that the required data will be available to support the function when needed. This is a cross-function integration requirement. The group must ensure that a process to create the attribute either already exists or will be established in this or some other business function.

STEP 3.2.7 COMPOSITE FUNCTIONAL INFORMATION SYSTEM REQUIREMENTS

Purpose: To validate and document the composite functional information systems requirements.

Description: The functional information systems requirements describe what an information system must do to support the needs of the composite function. The requirements describe what will be accomplished, while the design performed by information systems professionals describes how the needs will be accomplished.

The group will validate the process and data models developed in Phase III. The group will then review the functional business plan from Step 2.1.9 to identify any additional policy, interface, and management requirements identified during the analysis. The process and data models and the additional policy, interface, and management requirements will be documented as the final list of functional information systems requirements.

Outputs:

- o Composite functional information systems requirements
    - o Validated process and data models
    - o Validated subprocess to entity (CRUD) matrix
    - o Validated composite scope, vision, and strategy
    - o List of additional policy, interface, and management requirements

Relationships: The functional information systems requirements will prioritized in the next step. The prioritized requirements are the basis for the information systems implementation strategy, the final output of this process.

Task 3.2.7.1 Validate Process and Data Models

Purpose: To package the process and data models as the list of information systems functional requirements.

Outputs:

- o Validated process and data models
- o Validated subprocess to entity (CRUD) matrix
- o Validated composite scope, vision, and strategy

Relationships: The information systems functional requirements are the primary input for the prioritized information systems requirements.

Approach: This task will be accomplished through four subtasks.

Subtask: Validate the Process Requirements from the Detailed Process Model

The group will list the subprocess documented during Step 3.2.4 in the order presented in the process to attribute matrix in Step 3.2.6. The group will compare the subprocesses with the the composite functional concept. The group will identify any elements of the composite functional concept not represented by the subprocesses and will determine whether additional subprocesses should be identified to provide information systems support for the composite vision. If additional subprocesses are required, the process model will be refined by the group. As part of this analysis, the scope, vision, and strategies will also be reviewed and updated if necessary. Following any changes, the group will examine the (revised) process model to ensure consistency of level, description or definition, and potential interfaces.

Subtask: Validate the Data Requirements from the Detailed Data Model

The group will list the entities developed during Step 3.2.5 in the order presented in the subprocess to attribute matrix in Step 3.2.6. The group will compare the entities with the composite functional concept. The group will identify any elements of the composite functional concept that are not represented by the entity-relationship diagrams and will determine whether additional entities or relationships should be identified to provide information systems support for the composite vision. If additional entities or relationships are required, the data model will be refined by the group. As part of this analysis, the scope, vision, and strategies will also be reviewed and updated if necessary. Following any changes, the group will examine the (revised) data model to ensure consistency of level, description or definition, and potential interfaces.

Subtask: Validate the Integration of the Data and Process Models

The group will analyze the modules completed in Step 3.2.4 and the subprocess to entity matrix in Step 3.2.6 to identify any subprocesses that may be more effectively completed by another function. The strategy for assessing these functions is to perform a series of analytic steps.

The group will first identify all entities that are not created by subprocesses in this function. The group will review the entities that are updated and deleted to determine whether the subprocesses performed in this function manage these entities. If the operations are very limited and performed by very few subprocesses, then the possible "outlier" (subprocesses that may belong in a different function) should be listed for further review by the CIM functional integration teams.

The group will then review the entities that are read only to ensure that the action diagrams represent the entity as an interface. If the action diagrams describe a different type of subprocess, then the subprocess to entity matrix should be re-evaluated to ensure that the proper coding of the operation exists (create, update, or delete). If the operation is still read only, then the subprocess should be evaluated in terms of whether it provides direct information systems support of the vision elements. If the subprocess is key to the support, then the group will document it as part of the scope of the functional requirements. If the subprocess is not key, then it should be identified as a possible outlier to be performed by a different function. The potential "re-location" of the subprocess is then documented and provided to the integration team. If the subprocess defines the read operation as an interface, then the group will document it separately and provide it to the integration team for further analysis and placement in the relevant function.

There may be some cases where the "ownership" of the subprocesses is difficult to determine based on the re-analysis of the subprocess to entity matrix. In such cases, the subprocesses will be identified and provided to the functional integration team for review.

Subtask: Prepare Introductions to Validated Process and Data Models

The group will write a brief introduction to the validated process and function models. The introduction will include the use of the models, their relationship to the design process, and how they will be updated and maintained. Any unresolved questions of process ownership are also listed in the introduction. The introduction will not try to restate the descriptions of process and data requirements but may include some themes if appropriate.

Task 3.2.7.2 Compile Additional Policy, Interface, and Management Requirements

Purpose: To compile the list of additional policy, interface, and management requirements.

Outputs:

- o List of additional policy, interface, and management requirements

Relationships: The additional requirements provide added information for the designers of the composite information systems.

Approach: This task will be accomplished through three subtasks:

Subtask: Review the Functional Business Plan for Possible Additional Requirements

In this subtask the group reviews the business analysis in Phase II to identify any additional systems requirements they feel are not included in the data and process models. As part of the analysis of the functional concept in Step 2.2.5 the group identified requirements for operational and organizational changes that may need to be supported by the information system. In addition, technical and architectural needs, interface needs, and additional management needs may have been identified during the analysis and compilation of the functional business plan. Examples of these systems requirements may include security, performance, user interface, reporting, accuracy, workload, and training.

The group lists these textual requirements for analysis in the next subtask. This listing is meant to simply generate known system and performance requirements. It is not a substitute for the further analysis required by the design team once the requirements are provided as part of the implementation strategy, Step 3.1.9.

Subtask: Document Additional Requirements

The group will examine the list of textual requirements identified in the previous subtask and identify categories for classifying these requirements. They may include organizational, technical and architectural needs, interface needs, management needs, and any other particular requirements noted. The group then will list the requirements in these categories. The requirements are then complete and can be added to the documentation of the validated models prepared in Task 3.2.7.1.

Task 3.2.7.3 Prepare the Functional Information Systems Requirements Document

Purpose: To assemble the functional information systems requirements into a single document.

Outputs:

- Composite functional information systems requirements

Relationships: The functional information systems requirements will be used in the requirements prioritization in the next step.

Approach: The outputs from Tasks 3.2.7.1 and 3.2.7.2 will be compiled into a single document and then submitted to the CIM integration teams for review.

STEP 3.3.1 EXISTING INFORMATION SYSTEMS HIGH-LEVEL PROCESS MODEL

Purpose: To develop a logical model of the processing within selected existing or planned information systems.

Description: In step 2.3.3 the group selected a few candidate information systems for supporting the function. In this step, the group will develop a high-level model describing the logical processing within each of those systems.

Because systems documentation, development methodologies, and other indicators of process vary widely, the specific techniques used to develop the high-level process model will differ. Therefore, the group will begin this step by reviewing system documentation, especially the design documentation, to determine what approach was used when the system was originally developed and during its last modification. Based on the review, the group will determine how to build the process model for the system and will then build it using the documentation approach defined for the future and composite functional requirements.

Outputs:

- o High-level information systems process model

Relationships: The information will be used in developing the Implementation Strategy, Step 3.1.9.

Task 3.3.1.1 Determine Process Modeling Approach

Purpose: To identify and evaluate the techniques for developing the high-level information systems process model.

Outputs:

- o Process analysis approach

Relationships: The approach is used in developing the high-level information systems process model.

Approach: This task will be accomplished through three subtasks

Subtask: Assess Systems Documentation and Design Methodology

The group will gather documentation supporting the system, to include design documentation, MAISRC-related documentation, and any manuals for using or maintaining the system. The documentation will be reviewed and a first level assessment of its completeness will be made. The group should look for the following types of indicators: documentation suite complying with DoD 7920.2-M, "Automated Information System Life-Cycle Management Manual," and DoD 7935A, "DoD Automated Information Systems (AIS) Documentation Standards"; program maintenance manual with instructions at the module level; CASE tool output as part of the manual; configuration management procedures applied to baselines; and design specifications documenting requirements traceability, interfaces, and special systems requirements.

Once the group is familiar with the documentation, the group may want to confer with the information systems program office or the information systems operators/maintainers to determine the currency of the documentation and the methodology used during the systems life cycle.

If the documentation is current, complete, and describes a structured methodology, the group may want to simply transfer the graphical presentation of the system into the notation used by this Process Guide. If the documentation is faulty, outmoded or incomplete, the group must determine whether there are means to overcome the faulty documentation to complete the high-level process model.

Subtask: Determine Techniques and Resources for Process Redesign

The group will review the findings from the previous subtask and determine if the documentation is sufficient for generating the high-level process model. If the process model can be described, then this subtask is complete. If the documentation is only partially sufficient, the group will identify the key gaps in the documentation and then identify specific techniques for determining the high-level process model. Some options include: looking at the business practices of the service or agency who maintains the system; interviewing the systems maintainers; examining the inputs and outputs of the system to determine the general processing capability; surveying maintainers, users, or managers; and/or reviewing the modules of the system. The group may determine that re-engineering tools should be used. The manual reviews and tools may require the support of additional technical experts or services. In addition, the group may request help from CIM technical experts in assessing their options for developing the high-level information systems process model.

Estimates of the resource requirements for using the techniques should be identified at this time. The resource requirements should be presented to the CIM director for review before a plan is completed in the next subtask.

Subtask: Plan the Process Redesign

In this subtask the group will complete a plan for use of the different techniques identified int he previous subtask. This plan will include a schedule for completion of different tasks and the resources required.

Task 3.3.1.2 Develop High-Level Information Systems Process Model

Purpose: To build a logical process model for the information system that will facilitate comparison with prioritized functional requirements.

Outputs:

- o High-level information systems process model

Relationships: The high-level information systems process model is compared with the prioritized information systems requirements.

Approach: During this task, the group will first review the results of the candidate information systems rankings performed in 2.3.3. The group will list the high-level functions of each system. The group will then apply the techniques identified in 3.3.1.1 to each information system to generate high-level action diagrams like those described in 3.1.1. These action diagrams will be grouped according to the functions from 2.3.3 and will be prepared for further analysis in Step 3.1.9.

STEP 3.3.2 EXISTING INFORMATION SYSTEMS HIGH-LEVEL DATA MODEL

Purpose: To develop a logical model of the data managed within selected existing or planned information systems so that the information systems' capabilities to meet functional requirements can be effectively compared.

Description: In step 2.3.3 the group selected a few candidate information systems for supporting the function. In this step, the group will develop a high-level model describing the logical data model within each of those systems.

Because systems documentation, development methodologies, and other indicators of data requirements vary widely, the specific techniques used to develop the high-level data model will differ. Therefore, the group will begin this step by reviewing system documentation, especially the design documentation, to determine what approach was used when the system was originally developed and during its last modification. Based on the review, the group will determine how to build the data model for the system and will then build it using the documentation approach defined for the future and composite functional requirements.

Outputs:

- High-level information systems data model

Relationships: The information will be used in developing the Implementation Strategy, Step 3.1.9.

Task 3.3.1.1 Determine Data Modeling Approach

Purpose: To identify and evaluate the techniques for developing the high-level information systems data model.

Outputs:

- o Data analysis approach

Relationships: The approach is used in developing the high-level information systems data model.

Approach: This task will be accomplished through three subtasks

Subtask: Assess Systems Documentation and Design Methodology

The group will gather documentation supporting the system, to include design documentation, MAISRC-related documentation, and any manuals for using or maintaining the system. The documentation will be reviewed and a first level assessment of its completeness will be made. The group should look for the following types of indicators: documentation suite complying with DoD 7920.2-M, "Automated Information System Life-Cycle Management Manual," and DoD 7935A, "DoD Automated Information Systems (AIS) Documentation Standards"; program maintenance manual with instructions at the module level; CASE tool output as part of the manual; configuration management procedures applied to baselines; and design specifications documenting requirements traceability, interfaces, and special systems requirements.

Once the group is familiar with the documentation, the group may want to confer with the information systems program office or the information systems operators/maintainers to determine the currency of the documentation and the methodology used during the systems life cycle.

If the documentation is current, complete, and describes a structured methodology, the group may want to simply transfer the graphical presentation of the system into the notation used by this Process Guide. If the documentation is faulty, outmoded or incomplete, the group must determine whether there are means to overcome the faulty documentation to complete the high-level data model.

Subtask: Determine Techniques and Resources for Data Model Redesign

The group will review the findings from the previous subtask and determine if the documentation is sufficient for generating the high-level data model. If the data model can be described, then this subtask is complete. If the documentation is only partially sufficient, the group will identify the key gaps in the documentation and then identify specific techniques for determining the high-level data model. Some options include: looking at the business practices of the service or agency who maintains the system; interviewing the systems maintainers; examining the data structures; surveying maintainers, users, or managers; and/or reviewing the modules of the system. The group may determine that re-engineering tools should be used. The manual reviews and tools may require the support of additional technical experts or services. In addition, the group may request help from CIM technical experts in assessing their options for developing the high-level information systems process model.

Estimates of the resource requirements for using the techniques should be identified at this time. The resource requirements should be presented to the CIM director for review before a plan is completed in the next subtask.

Subtask: Plan the Process Redesign

In this subtask the group will complete a plan for use of the different techniques identified int he previous subtask. This plan will include a schedule for completion of different tasks and the resources required.

Task 3.3.1.2 Develop High-Level Information Systems Data Model

Purpose: To build a logical data model for the information system that will facilitate comparison with prioritized functional requirements.

Outputs:

- o High-level information systems data model

Relationships: The high-level information systems data model is compared with the prioritized information systems requirements.

Approach: During this task, the group will first review the results of the candidate information systems rankings performed in 2.3.3. The group will list the high-level functions of each system. The group will then apply the techniques identified in 3.3.1.1 to each information system to generate high-level entity relationship diagrams like those described in 3.1.2. These action diagrams will be grouped according to the functions from 2.3.3 and will be prepared for further analysis in Step 3.1.9.

STEP 3.1.8 PRIORITIZE INFORMATION SYSTEM REQUIREMENTS

Purpose: To analyze the functional requirements, resolve any differences, and reorder the functional information systems requirements in priority sequence. The prioritization will be based primarily upon functional need and economics (i.e. cost/benefits), followed by technical feasibility and other factors such as safety and security.

Description: In previous Phase III steps future and composite functional information systems requirements were documented in the form of a process model. In this step the future and composite requirements will be compared. The resulting list of differences will document the relationship of the future and composite functional information systems requirements to each other. An evaluation of composite requirements differences (unmatched to future requirements) must be performed to determine their disposition. They may be functions which are accomplished manually, overlooked in the development of future requirements in previous steps, or obsolescent in the future function.

The functional information systems requirements developed thus far were unconstrained by environmental considerations such as requirement complexity and cost. However, a key assumption is that the future functional concept represents a better, more efficient, and more economical way of doing business. To facilitate the development of an effective and efficient information systems implementation strategy, priorities must be established in this step for each requirement. The group will review and update the detailed cost/benefit data from Step 2.1.9 for each functional information systems requirement and reorder the requirements in priority sequence. Prioritization will be primarily based upon the cost/benefit data, although there may be other factors which will influence prioritization. It is assumed that functional need will significantly influence the benefit half of the cost/benefit analysis.

Outputs:

- Matrix of differences between future and composite requirements
- Prioritized requirements

Relationships: The functional information systems requirements from Step 3.1.7 and 3.2.7, and the Cost/Benefit data from Step 2.1.9 will be used to develop the matrix of differences and to prioritize information systems requirements. The matrix of differences and prioritized information systems requirements will be used in Step 3.1.9 to develop an information systems implementation strategy and transition plan.

Task 3.1.8.1 Comparison of Future and Composite Functional Information Systems Requirements

Purpose: To compare the future and composite functional information systems requirements, develop a matrix of differences, and analyze those differences.

Outputs:

- o Subprocess differences matrix

Relationship: Future and composite functional information systems requirements were identified in Steps 3.1.7 and 3.2.7 respectively. The list of differences between the two sets of requirements will be prioritized in Task 3.1.8.2.

Approach: This task will be accomplished in two subtasks.

Subtask: Identify Differences

The future to composite requirements matrix will be developed in this task for the requirements identified in Steps 3.1.7 and 3.2.7. An action table matrix form is included in this guide to document these differences. (On the form the columns referring to selected systems will be used in the next step.) It is critical that this information be captured as accurately as possible.

The list of differences (matrix) should then be reordered to group the requirements by category. All future requirements which match composite requirements should be grouped together in one group, unmatched future requirements in a second group, and unmatched composite requirements in a third group.

Subtask: Analyze and Resolve Differences

Differences appearing on the matrix in the composite column (i.e., unmatched composite requirements) must be reviewed and resolved where necessary. If the difference is simply a naming convention mismatch, an oversight in documenting a future requirement, or a function accomplished manually, then make the appropriate corrections to the comparison matrix. If the difference is a business activity which is no longer required, then the requirement probably can be eliminated. If the difference is valid, then no action is required.

The future functional information systems requirements which are unmatched to composite requirements should not require a great deal of analysis. These requirements were developed from the future visions (Phase I) and future functional concept (Phase II) and represent new or changed functional information systems requirements.

Requirements to Selected Information Systems Matrix
Action Diagram Subprocesses

Requirements	Future	Composite	Selected Systems				
			1	2	3	4	5

Task 3.1.8.2 Prioritize Future and Composite Functional Information Systems Requirements

Purpose: To develop the criteria for prioritizing functional information systems requirements, apply the criteria to the functional requirements, and reorder the requirements in priority sequence.

Outputs:

- o Prioritization criteria
- o Prioritized requirements

Approach: This task will be accomplished through two subtasks.

Subtask: Develop Prioritization Criteria

The primary consideration in prioritizing the functional information systems requirements is that the target objective will be a better, more efficient, and more economical way of doing business within this business function. As each requirement is considered, the benefits (tangible and intangible) must be explicitly identified and quantified wherever possible. Similarly, the cost of implementing the requirement must be identified and quantified.

This is a very complex process and one that should be approached in a manner which will thoroughly consider the many categories of cost/benefit criteria which may be applicable. The functional information systems requirements developed in Phase II were subjected to a cost/benefit analysis in Step 2.1.9. This analysis must be reviewed and updated to build the "business case" for including the requirement in the future standard system. Specific things to consider in updating the cost/benefit information are as follows:

<u>Benefits</u>

Reduced resource requirements

- Personnel
- Lease, rental, and maintenance
- Support services
- Training
- Supplies and utilities

Improved operations

- Reduced error rates
- Improved information technology utilization

<u>Costs</u>

Operational and functional training
Hardware requirements
Telecommunication requirements
Security
Support software
Support services
Supplies
Lease, rental, and maintenance

Note: Information describing these costs and how to perform cost/benefit analysis is described in DoD Instruction 7041.3, "Economic Analysis and Program Evaluation for Resource Management," October 18, 1972.

There may be additional categories of benefits and costs based upon the functional information systems requirement itself. The functional group must identify these. Costs and benefits should be projected on an annual basis when they are recurring or continuous in nature.

After the cost/benefit data is developed for each functional information systems requirement, the requirements should be reordered (prioritized), generally ranking the requirement with the greatest benefit and least cost first, etc, with the least benefit, highest cost last.

There may be criteria for specific requirements that will affect the prioritization without considering cost/benefits. Such criteria may be mission criticality, technical feasibility, safety, health, security, external and internal interfaces, legislative mandates, regulatory requirements, mobilization, and customer satisfaction levels.

The prioritization criteria and the techniques for applying them should be documented for future reference.

After the functional information systems requirements have been prioritized they should be reordered in descending order beginning with the highest priority requirement and ending with the lowest priority.

STEP 3.1.9 IMPLEMENTATION STRATEGY

Purpose: To develop an implementation strategy to transition to a future functional standard information system.

Description: There may be several information systems existing which may or may not support the function. The information system strategy can range from (a) adoption of a current operational system, to (b) development of a totally new system derived from the future functional information systems requirements. The implementation strategy should provide continuous high-quality and cost-beneficial information systems support to the function. It should also define the short term, intermediate term, and long term transition plans to the future standard system.

In previous steps the group documented functional information systems requirements for the function, both future and composite. This information was then subjected to a comparative analysis which resulted in the identification of differences between future and composite requirements. The requirements were then prioritized, based primarily upon cost/benefit analyses and functional needs.

In this step, the prioritized requirements will be compared to each of the selected information systems models to serve as a base point for development of the implementation strategy. The matrices of differences developed in Step 3.1.8 will be completed and refined by identifying which of the selected information systems each of the functional requirements. Completion of this task will require supplementing the group with experts from central design activities.

The updated matrices will be used to develop an implementation strategy to transition to a future standard information system. Several transition alternatives for satisfying these requirements will be developed,one of which will be recommended for approval. The group will then develop goals, objectives, and an implementation strategy for providing continuous information systems support to the function while the future standard information system is being developed. The implementation strategy will be structured to be fully consistent with the Department's program management policies.

The Phase II Functional Business Plan (Step 2.1.9) resulted in a first iteration of the documents to be submitted to the MAISRC (Milestone 0) under Life Cycle Management guidance. This step will include an expansion of these documents to the point that all information required in the MAISRC (Milestone 0) System Decision Paper (SDP) will be satisfied. In fact nearly all of the documentation required for MAISRC, (Milestone 1) will be captured. The SDP includes the Mission Needs Statement, an Economic Analysis of each transition alternative, and the

Program Management charter.

Outputs:

- o Updated matrices of differences
- o Transition alternatives
- o Preferred transition alternative
- o Information systems goals and objectives
- o Implementation strategy
- o Transition plan
- o System Decision Paper
    - o Mission Needs Statement
    - o Economic analysis
    - o Program management charter

Actual implementation of the strategy is outside the scope of the Process Guide.

Relationships: This step is the culmination of the CIM process.

Task 3.1.9.1 Identify Differences between Prioritized Requirements and Selected Information Systems Models

Purpose: To compare the prioritized functional information systems requirements to the selected information systems and identify the requirements which are currently being satisfied by one or more of the systems.

Outputs:

- o Completed matrices of differences

Relationship: The updated matrices of differences (action tables and data entity) will be used to develop transition alternatives and an implementation strategy.

Approach: The prioritized functional requirements from Task 3.1.8.3 will be compared with the high-level information system process and data models for each selected information system. The action table and data entity matrices from Task 3.1.8.2 will be annotated for each selected information system, if the prioritized requirement is satisfied by that system. A comparison will be made for each selected information system.

As the requirement to system comparison proceeds, the group may identify capabilities provided by the selected systems which are not documented as functional requirements. If discovered, these "unmatched existing systems requirements should be added to the appropriate matrix.

There are two significant results of this comparison. The first will be visibility of the capability of each selected system to satisfy the functional information systems requirements. The second will be identification of functional requirements which are not satisfied by any of the selected systems. The visibility resulting from this comparison will be critical to the development of transition alternatives.

Task 3.1.9.2 Identify Alternative Transition Strategies

Purpose: To develop alternative transition strategies for providing continuous information systems support to the function as we transition to the future standard system.

Outputs:

- o Transition alternatives

Relationships: The Phase II and Phase III process thus far has provided a number of documents. A Functional Business Plan was developed in Step 2.1.9. Detailed process models and data models were integrated in Step 3.1.6. Aggregated future and composite functional requirements were compared to each other in Step 3.1.8, and to the selected information systems Task 3.1.9.1. The functional requirements for the function were prioritized based upon cost/benefit analyses and functional need, and a list of prioritized requirements developed in Task 3.1.8.2. All of these documents will be used to develop transition alternatives to the future standard system. These transition alternatives will be the basis for an Economic Analysis to be performed in Task 3.1.9.3 and the identification of a preferred transition strategy in Task 3.1.9.4.

Approach: In this task the differences between the capabilities provided by the selected information systems and the prioritized information systems requirements will be analyzed to determine transition alternatives. The alternatives may be short, intermediate, or long term in nature, depending upon the disparity between the selected information systems capabilities and the prioritized information systems requirements. For example, in the best case situation, where there are few differences, an alternative may be developed to enhance a selected system immediately and field it as the standard system. In a worst case situation, where there are many differences, an alternative may be to develop a completely new system to be fielded as the DoD standard system. Most probably, the "real world" situation will fall between the best and worst cases.

This task will be accomplished in four subtasks.

Subtask: Analyze matrices of differences

The action table and data entity matrices must be analyzed to determine the capabilities of the selected information systems to satisfy the prioritized functional requirements. The analysis should consider the scope and complexity of each requirement as well as the number of requirements satisfied. If a requirement is satisfied by a selected information system and its scope and complexity is significant, it may of itself support consideration of the system as an alternative.

This analysis could be very brief if a selected information system satisfies only a few requirements. A decision could likely be made to eliminate that system from future consideration as an alternative. It is recommended that some threshold criteria be established below which a system is eliminated from further consideration.

Subtask: Compare selected information systems

Significant screening of potential information systems was done in Phase II to reduce the number of selected information systems to a few systems which are the most promising. This subtask uses information from Steps 2.3.3, 3.3.1, and 3.3.2 to compare selected information systems from a portability perspective. Significant system redesign or acquisition requirements may eliminate one or all of the selected information systems from further consideration as a standard interim system.

Completion of this subtask will require supplementing the group with experts from a central design activity.

Subtask: Identify alternative transition strategies

The previous subtasks may have eliminated some of the selected systems from further consideration in the development of alternative transition strategies. The remaining systems and the prioritized requirements will now be used to identify alternative transition strategies. One of the alternatives identified should be to continue to use existing systems (status quo). The reason for this is to baseline costs/benefits for considering other alternatives when the economic analysis is prepared and analyzed in Task 3.1.9.3.

Economic constraints and other factors such as data systems and telecommunications capabilities and capacities, may limit the number of prioritized requirements which can be accommodated by a given alternative. In this event the group may need to identify those requirements.

Completion of this subtask will require supplementing the group with experts from a central design activity.

Subtask: Summarize transition alternative information

The information gathered in the previous subtasks must now be summarized in narrative form for each transition alternative identified.

Task 3.1.9.3 Prepare the Economic Analysis

Purpose: To perform an economic analysis for each transition alternative identified in Task 3.1.9.2.

Outputs:

- o Economic analysis

Relationships: This economic analysis will be used in Task 3.1.9.4 to identify the preferred transition alternative. It will also be used in Task 3.1.9.6 to develop an implementation strategy and be incorporated into the System Decision Paper for MAISRC (Milestone 0).

Approach: A complete economic analysis will be prepared for each transition alternative. DoD Instruction 7041.3, "Economic Analysis and Program Evaluation for Resource Management," October 18, 1972, provides detailed guidance in the preparation of an Economic Analysis.

Task 3.1.9.4 Select Preferred Transition Alterative

Purpose: To analyze the alternative transition strategies and the economic analysis and select the preferred alternative.

Outputs:

- o Preferred transition alternative

Relationships: This analysis uses the list of alternative transition strategies developed in Task 3.1.9.2 and the economic analysis prepared in Task 3.1.9.3.

Approach: A complete in-depth analysis of the data gathered and organized in Task 3.1.9.2 and the economic analysis prepared in Task 3.1.9.3 will be performed for each alternative. Things to consider in this analysis are: satisfied goals and objectives, cost (acquisition, training, etc), benefits (resource savings, dollar savings), transition lead time, system redesign requirements, and payback or return on investment. Comparison of the data for these alternatives will result in a recommendation to pursue a specific transition alternative. A decision support software tool could be used to assist in evaluating the alternatives.

Task 3.1.9.5 Identify Goals and Objectives

Purpose: To identify goals and objectives to transition to the information systems support for the function.

Output:

- o Information systems goals and objectives

Relationships: The recommended alternative from Task 3.1.9.4 and the techniques used for developing Phase II goals and objectives (Steps 2.1.2 and 2.1.3) will be used to develop the Phase III information systems goals and objectives. These will be used in Task 3.1.9.6 to develop the implementation strategy and transition plan.

Approach: The goals and objectives developed in Phase II will be revalidated and updated in this task. The Phase III process has resulted in prioritized requirements and selection of a recommended transition alternative to the standard system. At this time the alternative must be compared to the Phase II goals and objectives to determine (1) the need to update goals and objectives and (2) the capability of the selected transition alternative to satisfy the goals and objectives.

Critical success factors and associated criteria for measuring success in achieving the goal must be established. If a critical success factor is not met, a goal will not be successfully completed. A threshold must be established that signifies success or failure to meet the goal.

Task 3.1.9.6 Develop Implementation Strategy and Transition Plan

Purpose: To develop the implementation strategy and transition plan for providing continuous, cost beneficial information systems support to the function.

Outputs:

- o Implementation strategy
- o Transition plan
- o System Decision Paper (SDP)

Relationships: The inputs to this task are the Phase III products developed in previous steps and the Phase II Functional Business Plan from Step 2.1.9. Outputs from this task will be used to transition to the standard information system and to obtain MAISRC approval (Milestone 0) approval to proceed.

Approach: The implementation strategy and transition plan must be structured in such a way as to be fully consistent with the Department's program management to facilitate implementation of the strategy. This will require establishing a charter for the project manager and actions to assure compliance with appropriate Departmental directives on automated life-cycle management. The final subtask is to update and restructure the Functional Business Plan from Step 2.1.9 to produce the System Decision Paper required by DoD 7920.2-M, "Automated Information System Life-Cycle Management Manual," for MAISRC (Milestone 0) reporting. The SDP includes the Mission Needs Statement, the economic analysis, and the program manager charter.

Subtask: Prepare Implementation Strategy and Transition Plan

As the program evolves beyond Phase III, internal documentation will be developed that demonstrates the level of thinking, analysis, and planning put into the program. This documentation addresses the topics normally considered in a well managed program and illustrates the level of planning for future phases of the program. Information to be included in the implementation strategy should include the following:

- Goals and objectives
- Establishment of a program manager (PM) and a PM charter
- Discussion of the transition alternatives considered and the alternative selected
- Integration management
- Functional and architecture strategy
- Most recent budget data
- Economic analysis
- An assessment of risk associated with the program and

how it will be managed
- Management of the schedule and milestones

Subtask: Develop Transition Plan

The transition plan provides plans for all the actions necessary to meet the information systems goals and objectives, recognizing that some of these plans will be developed during the transition process. It includes the plans for supporting:

- Transition
- Security
- Contingency
- Quality control
- Validation and verification
- Interface requirements
- Standards and interoperability
- Alternate designs and selection
- Model and simulation considerations
- Pilot processing
- Site preparation
- Procurement
- Software conversion
- Communications
- Deployment and implementation
- Configuration management
- Post deployment
- Maintenance and deployment of future functional requirements

Subtask: Develop System Decision Paper

This document serves two purposes for Milestone 0. It identifies and validates needs as expressed by functional requirements and recommends the exploration of alternative functional concepts. It includes the Mission Need Statement and the Program Manager charter. It should answer the question of "What do we want?" Specific concerns to be addressed at this point involve:

- Quantifying the identified mission deficiencies and the goals for improvement
- Describing the current and projected environment
- Estimating overall costs
- Determining affordability constraints
- Describing the needs with clarity and focus
- Determining what needs can be satisfied within current capabilities
- Establishing need priorities
- Determining the timing and urgency of the needs

The Functional Business Plan Documented in Step 2.1.9 probably satisfies most of the reporting requirements of the SDP. However, some of the requirements will be produced for the first time in Phase III and the entire document must be updated based upon phase III products. The sections of the MNS are worded slightly different than those of the Functional Business Plan. The wording should be revised to align with the MNS wherever possible.

FUNCTIONAL STEERING COMMITTEE MEMBERSHIP
FINANCIAL MANAGEMENT

<u>CHAIRMAN</u>:

Mr. Sean O'Keefe, DoD Comptroller

<u>EXECUTIVE SECRETARY</u>:

Mr. Nelson Toye, Director, Accounting Policy

<u>DOD COMPTROLLER</u>:

Mr. Alvin Tucker, Deputy Comptroller (Management Systems)
Ms. Cynthia Kendall, Deputy Comptroller (Information Resources Management)
Mr. Al Conte, Director, Consolidation Implementation Group (Defense Accounting and Finance Organization)
Ms. Belkis Leong-Hong, Director, Corporate Information Management

<u>ARMY</u>:

Mr. Douglas Brook, Assistant Secretary (Financial Management)
Mr. Neil Ginnetti, Deputy Assistant Secretary (Financial Management)

<u>NAVY</u>:

Mr. Robert McCormack, Assistant Secretary (Financial Management)
Capt. John Kavanaugh, Assistant Comptroller, Financial Management Systems/Commander, Navy Accounting and Finance Center

<u>AIR FORCE</u>:

Mr. Michael Donely, Assistant Secretary (Financial & Comptroller)
Mr. John Nethrey, Deputy Assistant Secretary (Plans, Systems & Analysis)

<u>DEFENSE LOGISTICS AGENCY</u>:

Mr. Richard Connelly, Comptroller
Mr. Charles Coffee, Director, Accounting and Finance Division

<u>WASHINGTON HEADQUARTERS SERVICES</u>:

Mr. David O. Cooke, Director
Mr. Joseph Friedl, Deputy Director for Budget and Finance

FUNCTIONAL STEERING COMMITTEE MEMBERSHIP
LOGISTICS

CHAIRMAN:

Ms. Diane Morales, DoD Deputy Assistant Secretary (Logistics)

EXECUTIVE SECRETARY:

Mr. James Reay, Director, Supply Management Policy, DASD(Logistics)

DOD COMPTROLLER:

Mr. Sean O'Keefe, Comptroller
Mr. Donald Shycoff, Principal Deputy Comptroller
Ms. Cynthia Kendall, Deputy Comptroller (Information Resources Management)
Ms. Belkis Leong-Hong, Director, Corporate Information Management

ARMY:

Mr. Eric Orsini, DoD Deputy Assistant Secretary (Logistics)
MGen. James Ball, Director of Supply and Maintenance, Deputy Chief of Staff (Logistics)

NAVY:

RAdm. Robert Moore, Assistant Commander for Inventory and Systems Integrity, Naval Supply Systems Command
RAdm. Edward Straw, Director of Material, Chief of Naval Operations

AIR FORCE:

BGen. Billy Barrett, Director, Logistics, Plans & Policy, Deputy Chief of Staff (Logistics & Engineering)

DEFENSE LOGISTICS AGENCY:

RAdm. J.P. Davidson, Executive Director of Supply Operations

MARINE CORPS:

Mr. Robert Riggs, Special Assistant to the Assistant Deputy Chief of Staff (Installation and Production)

FUNCTIONAL STEERING COMMITTEE MEMBERSHIP
MEDICAL

CHAIRMAN:

Dr. Enrique Mendez Jr., Assistant Secretary of Defense (Health Affairs)

EXECUTIVE SECRETARY:

Mr. Vance Kauzlarich, CIM Medical Deputy Functional Group Leader

DOD:

Mr. Sean O'Keefe, Comptroller
Mr. Christopher Jehn, Assistant Secretary (Force Management & Personnel)
Mr. Stephen M. Duncan, Assistant Secretary (Reserve Affairs)
Mr. Donald Shycoff, Principal Deputy Comptroller
Ms. Cynthia Kendall, Deputy Comptroller (Information Resources Management)
Ms. Belkis Leong-Hong, Director, Corporate Information Management
Dr. Michael Mestrovich, CIM Medical Functional Group Leader

ARMY:

Mr. G. Kim Wincup, Assistant Secretary (Manpower & Reserve Affairs)
LGen. F.F. Ledford, Surgeon General

NAVY:

Ms. Barbara S. Pope, Assistant Secretary (Manpower & Reserve Affairs)
VAdm. J.A. Zimble, Director of Naval Medicine/Surgeon General

AIR FORCE:

Mr. Jerome G. Cooper, Assistant Secretary (Manpower, Reserve Affairs, Installation & Environment)
LGen. Monte B. Miller, Surgeon General

JOINT CHIEFS:

RAdm. Joseph R. Smyth, Deputy Director for Medical Readiness

FUNCTIONAL STEERING COMMITTEE MEMBERSHIP
HUMAN RESOURCES MANAGEMENT

CHAIRMAN:

Mr. Christopher Jehn, DoD Assistant Secretary (Force Management & Personnel)

EXECUTIVE SECRETARY:

Ms. Norma St. Claire, DoD Force Management & Personnel

DOD:

Mr. Sean O'Keefe, Comptroller
Dr. Enrique Mendez Jr., Assistant Secretary of Defense (Health Affairs)
Mr. Stephen M. Duncan, Assistant Secretary of Defense (Reserve Affairs)
Mr. Donald Shycoff, Principal Deputy Comptroller
Ms. Cynthia Kendall, Deputy Comptroller (IRM)
Ms. Belkis Leong-Hong, Director, Corporate Information Management
Mr. Kenneth Scheflen, CIM HRM Functional Group Leader

ARMY:

Mr. G. Kim Wincup, Assistant Secretary (Manpower & Reserve Affairs)
LGen. A.K. Ono, Deputy Chief of Staff for Personnel

NAVY:

Ms. Barbara S. Pope, Assistant Secretary (Manpower & Reserve Affairs)
VAdm. J.M. Boorda, Deputy CNO (Manpower, Personnel & Training)

AIR FORCE:

Mr. Jerome G. Cooper, Assistant Secretary (Manpower, Reserve Affairs, Installation & Environment)
LGen. Thomas J. Hickey, Deputy Chief of Staff, Personnel

MARINE CORPS:

LGen. N.H. Smith, Deputy Chief of Staff (Manpower & Reserve Affairs)

DEFENSE CONTRACT AUDIT AGENCY:

Mr. William H. Reed, Director

DEFENSE LOGISTICS AGENCY:

LGen. Charles McCausland, Director

DEFENSE MAPPING AGENCY:

MGen. R.F. Durkin, Director

WASHINGTON HEADQUARTERS SERVICES:

Mr. David O. Cooke, Director